A MATHEMATICAL FRAMEWORK FOR IMAGE ANALYSIS

FINAL TECHNICAL REPORT

Co-Principal Investigators:

Donald Geman (University of Massachusetts—Subcontract)

Stuart Geman (Brown University)

Basilis Gidas (Brown University)

Ulf Grenander (Brown University)

Donald E. McClure (Brown University)

August 1991

OFFICE OF NAVAL RESEARCH

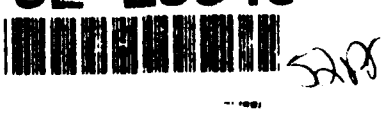
MATHEMATICAL SCIENCES DIVISION

CONTRACT NUMBER N00014-88-K-0289

Division of Applied Mathematics

Brown University

Providence, Rhode Island 02912



A MATHEMATICAL FRAMEWORK FOR IMAGE ANALYSIS

ABSTRACT. The results reported here were derived from the research project "A Mathematical Framework for Image Analysis" supported by the Office of Naval Research, contract N00014-88-K-0289 to Brown University. A common theme for the work reported is the use of probabilistic methods for problems in image analysis and image reconstruction. Five areas of research are described: rigid body recognition using a decision tree/combinatorial approach; nonrigid body recognition using deformable templates; the construction of two and three dimensional shape models for complex recognition and interpretation tasks; three dimensional shape reconstruction; and texture analysis.

1. INTRODUCTION

Our recent work has focused on "high level" computer vision. The mathematical models in this context are generally global ones (designed to embody global regularities), in contrast to lattice-based random field models (designed first to capture more local regularities), which pervaded much of our earlier work. Still, the global models build heavily on what we have learned from earlier research concerning, for example, the power and flexibility of probabilistic models built up from specifications of low-order, local (Markovian) dependence.

New mathematical issues arise in the formulation and analysis of high level computer vision problems. A central issue concerns *representation*—the association of an object with a mathematical model which incorporates essential invariant features of the object and which is flexible enough to represent different instances or presentations of the object. Another important issue concerns *decision procedures* for recognition—what are optimal strategies for making a set of measurements (tests) on an observed image and for using the outcomes to infer the presence or absence of an object and its classification.

Following this introduction, the report is organized into five sections (Sections 2 through 6). The next section, Section 2, discusses problems related to the recognition of rigid objects. We have recently implemented algorithms for character recognition in highly degraded environments and for 3D object recognition using range data. The implementation per se motivates mathematical questions of broader interest. In particular, we describe the use of relational templates for invariant rigid object representation, the use of so-called "interpretation guided segmentation" and its potential connection to statistical hypothesis testing, and relations between coding theory, sequential decision theory, and optimal decision procedures.

Section 3 introduces deformable templates for the modeling of nonrigid one-dimensional objects. This work has been guided by two application areas: recognition of coronary arteries in X-ray images of the heart (arteriograms) and handwritten character recognition. The one-dimensional nature of the models makes the use of dynamic programming feasible for the computation of solutions of global optimization problems. Formally, these deformable templates are examples of hidden Markov models, which have been much studied and successfully used in speech recognition. Part of the research described here concerned the theoretical issue of identifying the class of processes that can be approximated by hidden Markov models. The main result, that hidden Markov models are provably universal, is also discussed in Section 3.

Section 4 describes generic aspects of deformable template models and discusses how they have been and might be implemented for higher dimensional structures, such as surfaces and volumes in $\mathbb{R}^3$. This work has been largely motivated by the need for global models for structures such as biological objects, objects which at the same time exhibit common global regularities and significant variation from observation to observation. For example, a human heart has a well-determined global structure, but varies in form from one individual to another, or even for a single person, varies significantly in shape from one point in a cardiac cycle to another. One of the mathematical issues in this area concerns convergence to a desired equilibrium distribution of a temporal random process, modeled as a solution of a stochastic differential equation, for which either the state space has uncommon structure (e.g., for jump-diffusion processes) or for which paths through the state space are constrained by a particular algorithmic implementation.

Section 5 discusses the use of Markov random field (MRF) models for recovery of intermediate level structure in a scene. Motivating problems include (i) "shape from shading" for visible light images and (ii) construction of surface maps from synthetic aperture radar (SAR) imagery. These are inherently ill-posed problems where MRF models have shown promise as a systematic approach to regularization.

Finally, in Section 6 we discuss the application of MRF's, and closely related tools, to texture analysis.

2. RIGID BODY RECOGNITION AND COMBINATORICS

This section is composed as follows. We first provide a statement of the invariant rigid object recognition problem, together with examples of applications and a discussion of the important heuristic issues, such as *combinatorial complexity* and *focusing*. We then describe an algorithm, previously developed in the contexts of optical character recognition and three-dimensional object recognition with range data. This algorithm embodies a particular recognition strategy that we have explored from an applied and algorithmic viewpoint. Whereas it does addresses the central difficulties of the problem, it does so in a somewhat ad hoc manner, and we believe a satisfactory mathematical understanding is lacking. Consequently, we discuss in the final section (2.3) a theoretical framework, starting from "first principles" and mathematical ideas in coding theory, decision theory, and in nonparametric statistics, within which the algorithm above may be regarded as an approximation. Some of fundamental mathematical issues treated in §2.3 are the following:

(1) *Feature Selection.* This refers to the problem of choosing what functions of the image data will be used to make decisions, or equivalently, what are efficient *object representations.*

(2) *Optimal decision protocols.* This includes (i) Computing the best order in which to accumulate information (given a class of features) during the actual search, where, for example, performance is measured by the expected decision time. (We refer to this as the *Twenty Questions Problem* for reasons that will become clear later on.) This problem is at the heart of our mathematical investigation. (ii) Determining how to prune the list of active "hypotheses" as information is assembled, for example in a manner consistent with limits on type I errors (failed detections) and type II errors (spurious confirmations). Here, one must assume a model for object placement and for the image formation process, especially the noise statistics.

2.1 Problem Statement and Heuristics. Consider the following recognition problem, one that living organisms solve every day. We are given a list of 2D or 3D "objects"; for example, the letters of the alphabet, a collection of cars and trucks, or an assortment of machine parts or manual tools. The objects are regarded as rigid and represent *particular* instances of the given shape class; thus, for example, the letters are represented by a particular font, the vehicles are specific makes and models, and so forth. The objects are then arbitrarily positioned in 3-space (or in 2-space if they are two-dimensional, with respect to rotations and translations, and this "scene" is then imaged by an ordinary camera or perhaps by a range finding device. The scene may contain multiple objects, each in multiple

aspects; some objects may be partially occluded by others or by "clutter," such as foliage or other "background" objects. In addition, there may be "noise," artifacts, or other degrading effects caused by the way in which the scene was illuminated and sensed. The goal is then to construct a list of those objects present in the scene (or perhaps specify the locations at which objects occur) based on the image data and (exact) shape information about the objects, e.g., convenient analytic representations. This is the problem of *rigid body invariant object recognition.* It has been widely studied in the computer vision community and there are many papers related in one way or another to the algorithmic approach we describe below. Whereas we do not claim to perceive the problem in an entirely original fashion, the mathematical framework we present does appear to be new. In fact, to our knowledge, no convincing theoretical model exists either for the many heuristic search algorithms dedicated to special cases, such as optical character recognition (OCR), or for the nature of the biological solution.

We should emphasize the distinction between *rigid* and *nonrigid* (or *deformable*) objects, which are treated in other sections of this report. Evidently, the problem of nonrigid invariant object recognition is generally more difficult since, in addition to the multitude of representations induced by spatial positioning, there are the additional ambiguities associated with the varying intrinsic shapes of individual objects. (To appreciate the latter complexities, one has only to imagine the number of ways in which people write the letter "A" or the amount of variation within biological shapes such as leaves and hands.) In our problem the individual patterns exhibit no variability in shape and hence there is no probabilistic model for the shape classes. Still, the rigid body problem is quite challenging because we wish to solve it in *highly degraded environments*, including substantial degrees of noise, clutter, and variations in lighting.

There are two particular cases we shall use for illustrative purposes: they are representative of the problems encountered and of our practical experience in this area. The first is optical character recognition, in which the objects are the thirty-six alphanumeric characters and the images are ordinary visible light pictures obtained with a video camera. One specific application we have in mind is the automatic recognition of identification characters on silicon wafers from high magnification photographs; see Figure 2.1. The second case involves three-dimensional objects consisting basically of planar or near planar surfaces. To simulate this situation, we have constructed "objects" by randomly generated parallelepipeds and also constructed "scenes" and corresponding range images by randomly positioning these objects in space and computing the appropriate depth values from a reference point; see Figure 2.2 for one such example. Potential applications include industrial

robotics and inspection, and the automated recognition of military targets, such as ships and tanks in infrared and laser radar imagery.

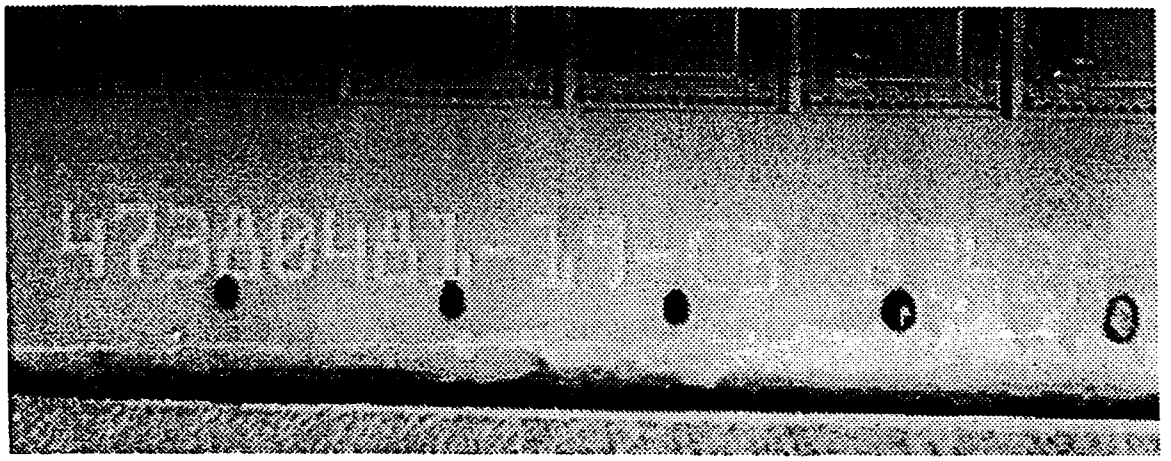


Figure 2.1. Silicon Wafer ID

The methodology we are exploring proceeds from the following assumptions and observations:

1. The most conceptually simple and straightforward approach would be to store a library of exact representations of all objects in all potential aspects and then search *individually* for these object-aspect combinations by some form of template matching. For example, given a reliable detection algorithm, the positions of potential objects could be located and, in principle, the matching could be done very effectively with optical correlators. However, there are several problems with this "brute force" approach. First, the enormous number of potential object-aspect pairings ("signatures") might impose unacceptable processing times, even for optical correlators. Second, and more importantly, correlation severely degrades in the presence of noise and clutter, and particularly so when distinct objects in varying aspects may have nearly identical signatures. By its nature, correlation *uniformly* emphasizes all locations; in particular, there is no mechanism for *focusing on ambiguous areas*—those where confusions are likely to occur.

2. Any search procedure should then proceed on a coarse-to-fine basis, in which many possibilities or "hypotheses" are considered at the early stages, giving way in a controlled progression to increasingly narrow and more specific investigations. The separation of objects with nearly identical presentations should be delayed until the last stages of the search procedure. One natural protocol is then a coarse-to-fine *decision tree*.

3. It has long been speculated (in the computer vision community) that object recog-

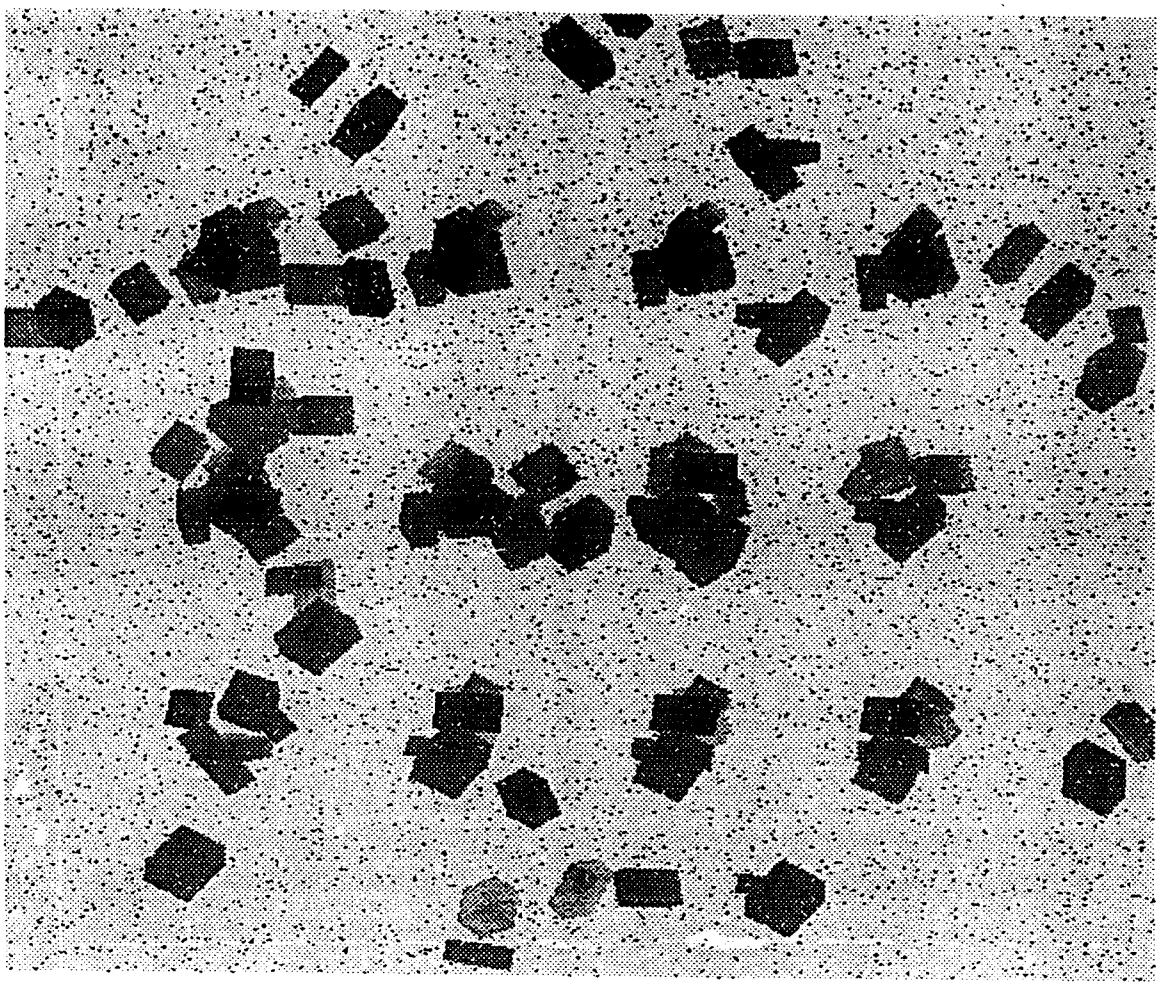


Figure 2.2. Simulated scene, twelve objects from two object types, plus clutter and noise

nition cannot be purely data driven and that decisions must in fact be "interpretation guided". In particular, we believe that any procedure based on blind segmentation, meaning fixed and universal thresholds, will fail in the presence of correlated noise, vagaries in illumination, and other factors common to real imagery. On the other hand, any approach involves extracting "features" from the data and representing these in a form sufficiently simple for comparison to stored representations, and this data conversion step necessitates "thresholding." It is precisely at this stage that we have found it effective to use *floating thresholds* by allowing pending classifications to determine the conversion parameters.

4. In conventional statistical classifiers, a collection of features (functions of the image data) is specified, and one attempts to estimate the conditional distribution of these features given the various hypotheses, the so-called class-conditional densities. The latter step is referred to as "training." Instead, the appropriate features could be *learned* and "training" could consist of an off-line (and perhaps intensive) optimization procedure in which the object representations are determined based on criteria such as economy and discriminating power. We refer to these elementary features comprising the object representations as "probes."

2.2 A Recognition Algorithm.

2.2.1 Overview. The basic idea is very simple. The algorithm involves sequentially visiting each (or most) image locations and implementing a decision tree for a field of view associated with that pixel. The output at each node of the decision tree is a list indicating which object-aspect pairings are "active" at the pixel, that is, have not been eliminated at any node. An object-aspect pairing is "true" within a field of view if the object is positioned there in such a way that a distinguished point in a subimage containing the ideal object-aspect signature is aligned with the origin of the field of view.

We will use the word *hypothesis* to indicate a particular object-aspect combination; thus, for example, in the OCR case there is one hypothesis for each planar rotation of each alphanumeric character and in the case of solid shapes, there is one hypothesis for each object type for each triple of angles corresponding to an appropriate sampling of azimuth, tilt, and rotations in a "ground" plane. The actual number and type of degrees of freedom allowed is problem dependent. Formally, at least, the only difference is in the number of hypotheses, which may be extremely large in real applications.

Information is extracted at predesignated offsets in the field of view subimage. In order to disambiguate hypotheses, this field must be larger than the minimum rectangle required to surround all hypothesis silhouettes. The hypotheses, or more precisely, the silhouette templates or *shapes*, are then mutually registered by aligning the centers of the rectangles circumscribing the silhouettes. This provides an origin for a reference coordinate system, and this origin may then be regarded as the image location at which we are attempting to detect and classify an object. In the ensuing discussion, image coordinates in the field of view are relative to this reference point.

Still fixing a field of view, the algorithm is based on a series of *probes* which are grouped into separate collections corresponding to nodes on a decision tree. These probes

refer to particular functions of image data (i.e., statistics or features) which are evaluated at predetermined locations, one type of function and one collection of locations for each node in the decision tree. Given a model for the image formation process (including object placement, noise, clutter, etc.), we may regard the probes as *random variables* indexed by sets of points or "offsets" and whose values are symbolic labels. These offsets are determined by an optimization procedure designed to minimize the error rates corresponding to false negatives (failed confirmations) and false positives (erroneous classifications). The functional form of the probes is determined in a more or less ad hoc fashion. We can also regard the probe values as "tests" upon which detection and recognition are based: the observed data values determine the action taken at each node of the tree. Hypotheses which are active at a given node and which "pass" a sufficient number of the tests for that node will remain active at the given location. The final output indicates which, if any, of the objects has been confirmed at the pending location; obviously most locations result in no confirmations.

The basic strategy is then a variant of "divide and conquer": many alternatives are pursued in parallel in the early stages, based on very general and mutually relevant criteria, whereas the intermediate stages focus on subclasses of hypotheses and finally, in the latter stages, the tests are designed to confirm or deny specific hypotheses against all the relevant alternatives, for example a particular orientation of the letter "E" against all pending aspects of other letters, usually ones similar (depending on the font) to "E", such as an "F".

2.2.2 Probe Selection and Optimization. Henceforth we concentrate on the problem of detecting 3D objects, such as those in Figure 2.1, based on range data. Exactly the same principles apply to intensity images and to other problems, such as OCR. The only information about the objects that will be utilized is the silhouette; in reality the images may be of sufficient resolution to provide useful information about the internal structure.

The purpose of the probes associated with the upper nodes of the decision tree is the rapid detection of a possible object at the given location. Consequently, these probes serve as "filters" to separate objects from background and to quickly eliminate most locations from further examination. Moreover, since in principle we allow no false negatives (unconfirmed objects), these filters must reliably identify those locations associated with actual objects. Consequently, most early confirmations are in fact false positives, i.e. do not correspond to a distinguished location on an actual object but result instead from object-like clutter or other objects at nearby locations. Specifically, for example, the probes in early nodes may then

simply be points (relative to our coordinate system) with a label which indicates whether the point should be "inside" or "outside" the *entire collection* of (registered) shapes. For simplicity, we might chose the same number J of inside and outside points, yielding $2J$ points in all and denoted by (I_j, O_j), $I_j = (I_j^1, I_j^2)$, $O_j = (O_j^1, O_j^2)$, $j = 1, 2, ..., J$. During the search, i.e. when the algorithm is actually executed, the image intensity (=range) values will be evaluated at these $2J$ points and these numbers will be correlated with the (binary) template values. Thus, ideally, the points must be chosen such that each I_j lies *inside* each shape and each O_j lies *outside* each shape. This can be accomplished by a relatively simple optimization procedure.

In contrast, the probes in the middle nodes of the tree should be designed to separate objects from clutter and from each other. Thus these nodes will typically be more computationally intensive although executed at only sparse locations, i.e. those which survive earlier decision nodes. They involve more complex probes since at this stage of the decision tree we now wish to compare many hypotheses simultaneously and retain those with some reasonable probability of occurrence at the current location. Remember that, ideally, a hypothesis is confirmed at this location exactly when the offset is zero. Since we are no longer primarily interested in separating objects from background, and since there are as yet no specific hypotheses to entertain, we desire probes which effectively disambiguate among all relevant pairs of hypotheses. Differences among all presentations of distinct objects must be precisely identified and exploited in a manner which is robust to noise, clutter, and parameter selection. This is the true *recognition* aspect of the problem.

We have experimented with probes which involve *relational template matching.* For example, we might associate with each *pair* of locations, usually in close proximity, a binary label corresponding to whether or not the pair of points straddles the object boundary, i.e. has a figure/ground relationship. The figure/ground dichotomy upon which the early probes were based is replaced by that of transition/no transition. In a real image containing that object, the transition pairs should typically correspond to significant differences in depth values whereas others should correspond to relatively small differences (depending on the nature of the "background"). Each hypothesis is again represented by a binary string and the object silhouette is recovered in the limit as the number of pairs increases.

It is important to notice that relational template matching necessitates that the actual intensity values that are extracted at the predetermined locations associated with a probe must be converted to a label, usually just 0 or 1, for comparison with the stored models. *A critical factor in the success of this approach is that threshold values used in the conversion of range values to labels be driven by pending interpretations.* The alternative,

using global thresholds, renders the algorithm unduly sensitive to parameter selection, illumination changes, and other factors, and results in unacceptable error rates. One method of incorporating this "top down" component is to use "floating thresholds"; see §2.3.7.

Specifically, for example, each probe might be a labeled *pair* (u, v) of locations. The label depends on which template or offsetted template is present at the reference point (i.e. within the field of view) and indicates the positioning of probe coordinates relative to the silhouette. Specifically, the label of (u, v) for hypothesis l is denoted $(I,O),(O,I)$ or (I,I) according to whether u is inside and v is outside shape l, vice versa, or both u and v are inside; we do not consider pairs for which *both* points lie outside any one of the templates. Now given two shapes, say l and k, with l at offset 0 and k at offset b (a vector) relative to the origin of the field of view and given a set x of N pairs of points, $x = (u_n, v_n), n = 1, 2, ..., N$, we define the discrepancy $D(x; l, k, b)$ between l and k as the number of indices n for which (u_n, v_n) has label (I,O) for shape l and label (O,I) or (I,I) for shape k, or label (O,I) for shape l and and label (I,O) or (I,I) for shape k. The rationale is that when we are examining the actual grey level (depth) image, we expect to find a smaller absolute depth difference $|u - v|$ between two (nearby) points u and v which are *both inside* an object than between two points, only *one* of which lies inside and the other in the "background". Moreover, we expect a point inside an object to be closer to the viewpoint than a point outside that object, which is why, since we also record the *sign* of $u - v$, we "credit" an $(I,O),(O,I)$ pair in determining D.

One way in which to choose the probe locations is to minimize a cost functional designed to separate all pairs of hypotheses. Fix $x = \{(u_n, v_n), n = 1, ..., N\}$ and define

$$H(x) = \sum_{(l,k,b)} ((\theta - D(x; l, k, b))^+)^2$$

where the sum extends over all pairs of shapes l and k, or perhaps only those associated with distinct objects, and all offset vectors b in some vicinity of the origin. We could then use coordinate-wise descent to find a value of x^* for which $H(x^*)$ is small, thereby providing a set of probes which separates (to the extent determined by the threshold θ) the presentations of all objects from those of all others. Notice that $H = 0$ if and only if every relevant pair l, k is at least θ units apart relative to D. When a field of view is fixed and the search is performed, the image intensity values are observed at the coordinates in x^* and for each hypothesis l the observed intensity differences for the pairs (u_n, v_n) are assigned one of the labels $(I,O),(O,I)$ or (I,I) using thresholds determined by the ideal proportions of label types for shape l (at offset zero); see §2.3.7. This is the hypothesis-driven segmentation we mentioned earlier. The result of this stage of the search is a collection of specific hypotheses

for which the Hamming distance between the observed and template values falls below a specified level.

Continuing on, we might then design hypothesis-specific tests in order to separately confirm or deny each of the now active specific hypotheses. Thus, for each hypothesis l, we might define select probes by minimizing a functional of the form

$$H_l(x) = \sum_{k,b}((\theta_1 - D(x;l,k,b))^+)^2$$

where the sum now ranges over offsets b and all hypotheses k corresponding to objects other than that associated with hypothesis l. At this stage probes should (and do) characterize subtle differences among the hypotheses.

The purpose of the final stages of the tree might be to utilize still more dedicated probes to disambiguate among confirmations which lie in close proximity. Or perhaps to exploit the *internal structure* of the objects, that is the depth differences among locations within the silhouette. Whereas computationally expensive, the procedure only occurs at the very bottom of the decision tree, and hence is only performed at very sparse locations and for candidate objects which have already "passed" all previous tests; consequently, the overall "cost" is no greater than that for the previous stages.

2.2.3 Experimentation. We have included four figures illustrating some preliminary experiments. Figures 2.3 and 2.4 show presentations of two "objects", seventy-two in all, corresponding to each object rotated every ten degrees in a fixed plane. Figure 2.5 shows an image constructed by randomly situating twelve such presentations, some of each type, together with clutter. There is also noise, obtained by replacing roughly 15 percent of the pixels by grey levels uniformly chosen from $0, 1, ..., 255$. Figure 2.6 shows the actual objects in the preceding figure correctly identified by an algorithm based on the principles outlined above. In addition, one of the clutter pieces was incorrectly identified as the object of which it is a subset; notice that the presented object, the actual object and the clutter are virtually indistinguishable.

2.3 Towards a Mathematical Framework.

2.3.1 Hypotheses, Tests, and the Twenty Questions Problem. Let us consider the rigid body recognition problem and attendant complexities in a somewhat more general fashion. We have a list of states of nature or *hypotheses*, which may represent spatial placements of rigid objects, as above, or perhaps phenomena of an entirely different character,

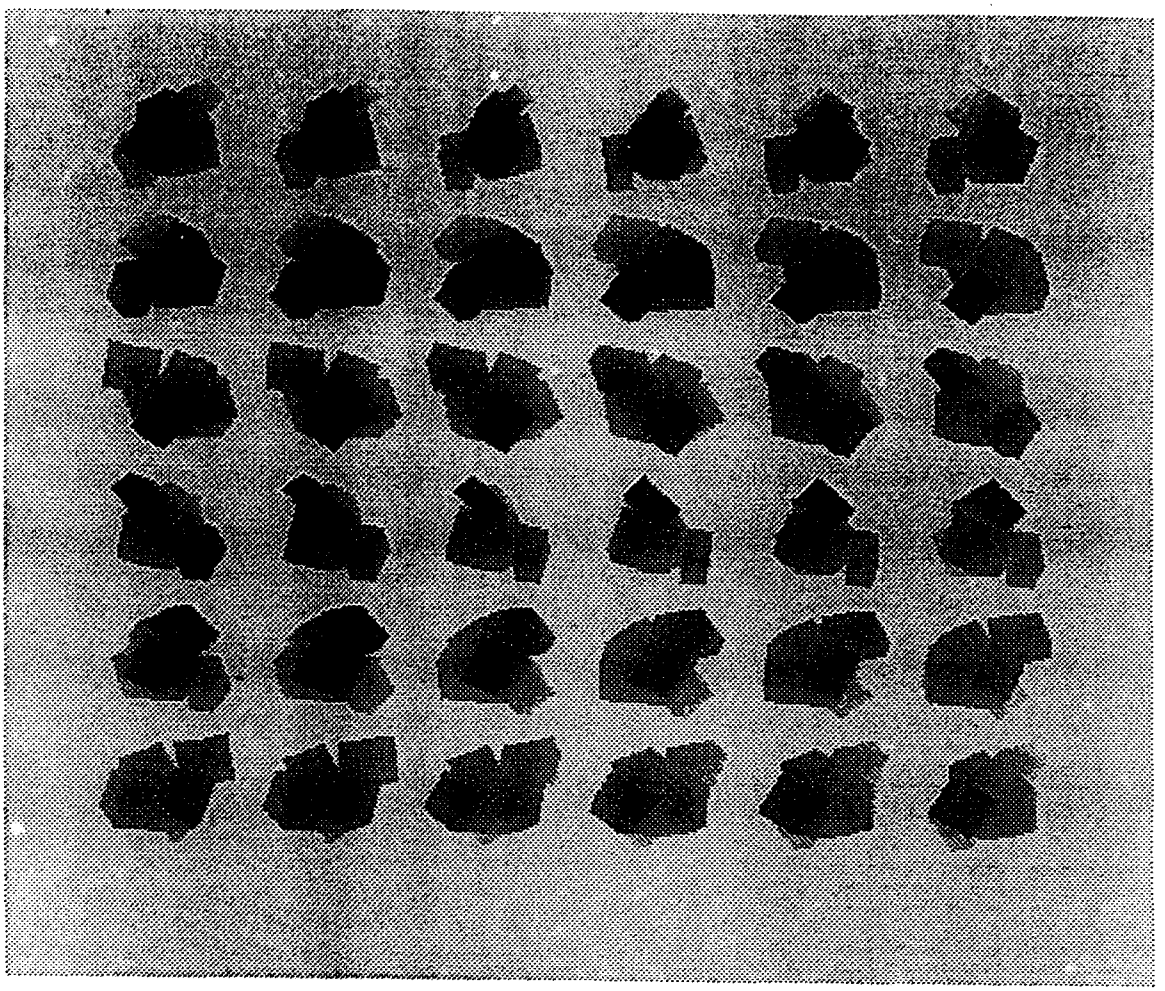


Figure 2.3. Object A, thirty-six aspects

such as possible diseases, causes for an accident, and so forth. In addition, we have *sources of information*, such as the "probes" we described above in the recognition problem, or such as specific medical tests, examinations, and patient histories in the medical diagnosis example. Let us continue to assume that information is available from a discrete collection of "tests", which, in view of the uncertainty about the true state of nature, may be regarded as random variables. In general, this family is extremely large and we may assume that no two hypotheses determine exactly the same test values. Our problem is then to perform these tests in an "optimal" order, say with respect to reaching a decision as soon as possible, and assuming the test choices are made adaptively, meaning that at each stage we may utilize the results of previous tests in order to choose the next one.

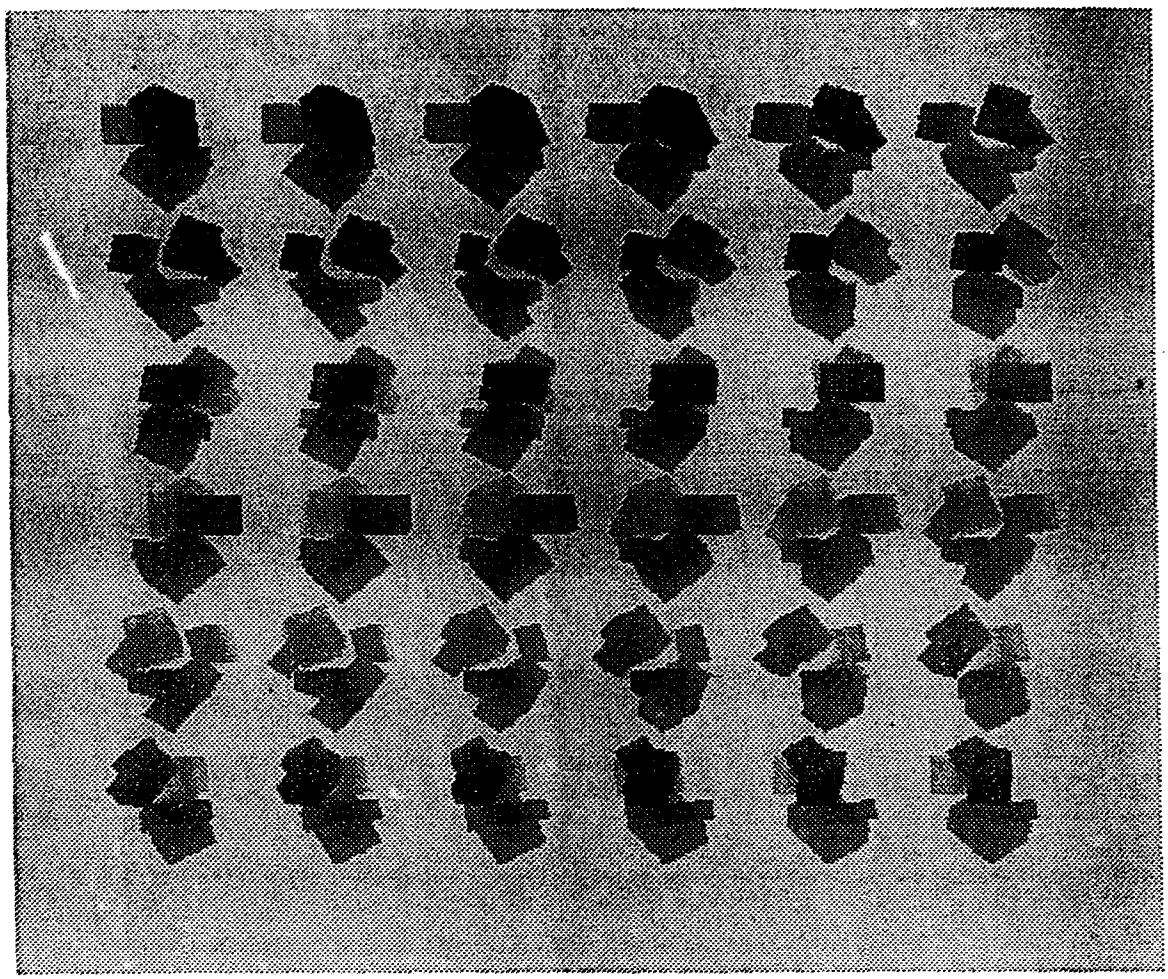


Figure 2.4. Object B, thirty-six aspects

Here is a specific example; it captures the essence of the problem and is known as the "Twenty Questions Problem", due to its resemblance to the once familiar parlor game and such old T.V. shows as "What's My Line?". One of two players chooses an entity m (famous person, historical event, unusual occupation, etc.) from among a known set of M entities according to a probability distribution $p = \{p_m, m = 1, ..., M\}$. The second player wishes to determine which entity was chosen and is allowed to select *any* subset $A \subset \{1, ..., M\}$ and ask the question "Is m in subset A?" Assuming the answers are truthful, what is the optimal strategy for minimizing the mean number of questions until the answer is known?

The solution is known, as well as bounds on the mean decision time, from results in

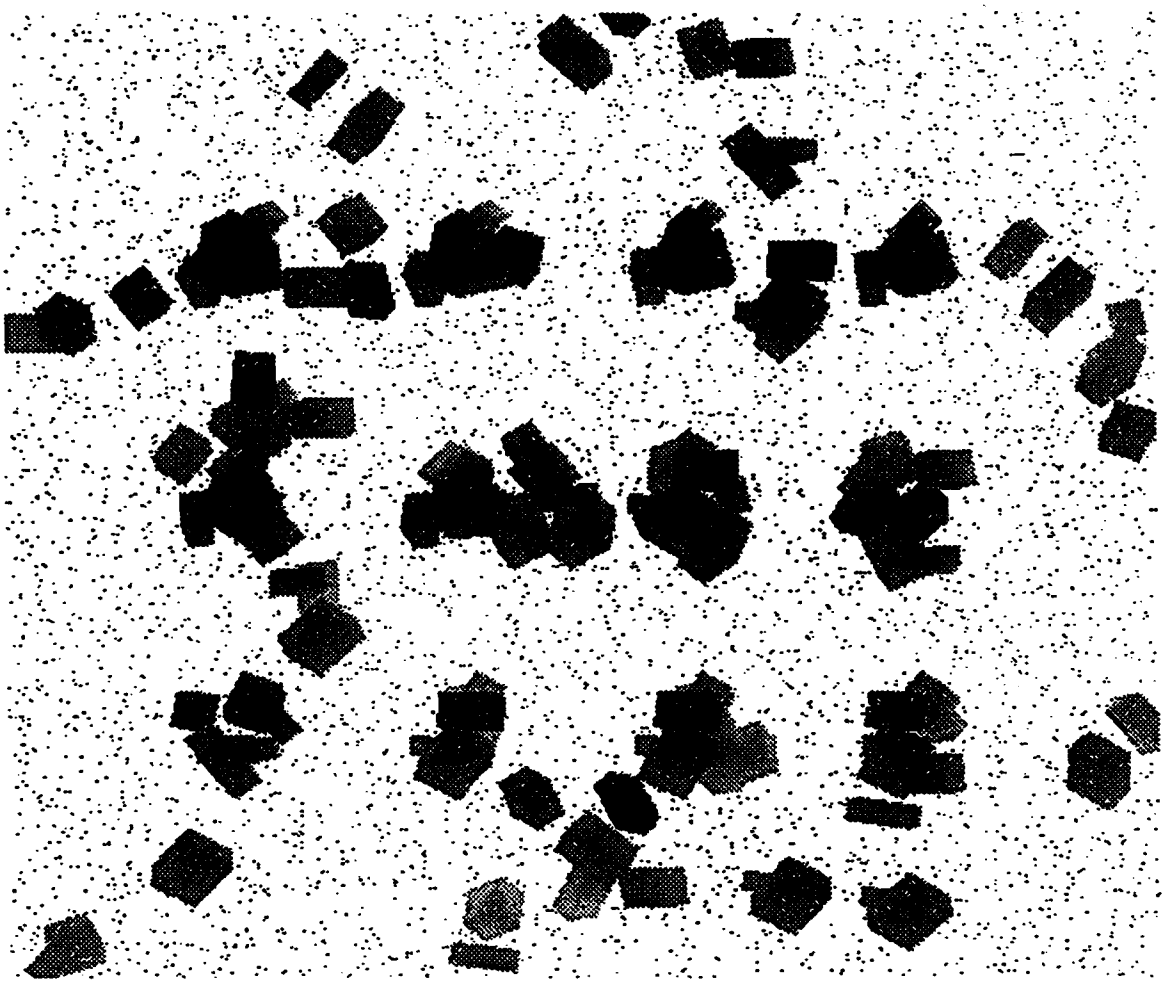


Figure 2.5. Simulated scene, twelve objects of types A and B, plus clutter and noise

coding theory. Specifically, given a set of symbols and probabilities $p = \{p_m\}$, the *Huffman code* [35] provides an explicit construction for the optimal way to code the symbols with (variable length) binary strings in the sense that the mean number of bits that must be examined during decoding is minimized. In the case of a binary code alphabet, the mean code length μ of the Huffman code satisfies the inequalities

$$H(p) \leq \mu < H(p) + 1$$

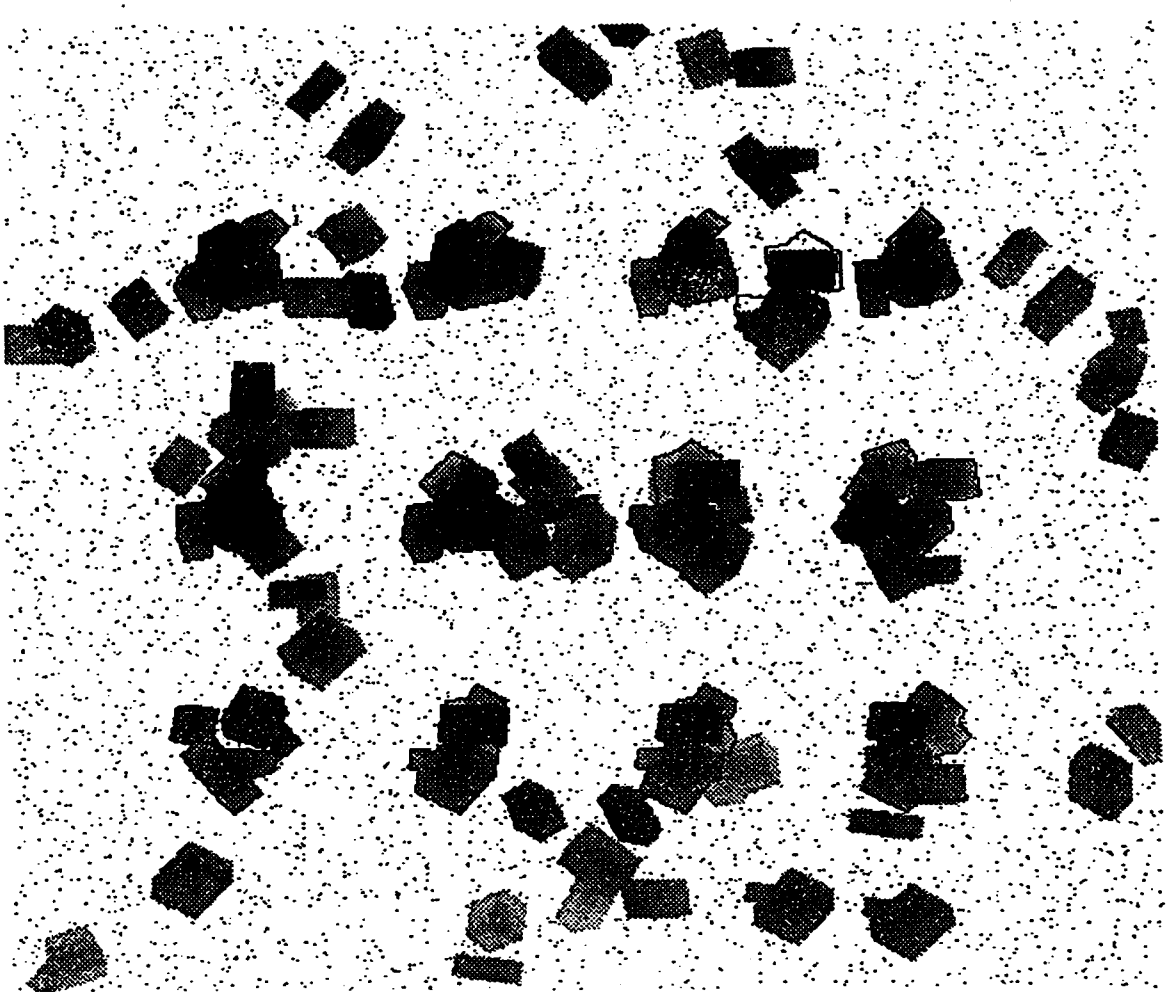


Figure 2.6. Objects in previous scene, all correctly identified

where $H(p)$ is the entropy of p:

$$H(p) = -\sum_i p_i \log_2 p_i.$$

Thus, for example, in our twenty questions problem, we would only need about twenty questions *on the average* if each of an original set of 2^{20} entities were equally likely to be chosen.

2.3.2 The Constrained Twenty Questions Problem. The Twenty Questions Problem is of course not truly representative of those we wish to solve in object recognition and related problems. For one thing, we wish to have tests (random variables) that are more

general than indicator functions. But more importantly, the problem is *too easy* as stated because, by allowing *all subsets*, the hypotheses are too well separated; in reality it may be impractical or impossible to have exactly the right question available at each instance, not to mention the problems encountered in actually storing or evaluating any collection of size 2^M when M is on the order of 1000, or even 30. For the time being let us ignore the former restriction but limit the set of possible questions:

Constrained Twenty Questions Problem. We are given a pair $(p, \mathcal{X})$ where $p = \{p_m, m = 1, \ldots, M\}$ is a probability vector and $\mathcal{X} = (X_{m,n}), m = 1, \ldots, M, n = 1, \ldots, N$, is an $M \times N$ *binary* matrix, i.e. $X_{m,n} = 0, 1$ for each m, n. There are then N "tests", represented by the columns of X in the sense that each column is associated in the obvious way with a subset of the "hypotheses" $\mathcal{M} = \{1, \ldots, M\}$. In addition, we assume the rows of $\mathcal{X}$ are distinct, so that the set of tests uniquely determines the hypothesis, and that the columns are also distinct, meaning that no test is "repeated." Given that a hypothesis is chosen according to p, the problem is then to determine the optimal strategy for performing the tests in order to minimize $E(\tau)$, the mean decision time. (Here, of course, strategies are adaptive; finding the *fixed* permutation of $\{1, \ldots, M\}$ which minimizes $E(\tau)$ is much easier, although still highly nontrivial.)

It is known [38] that this problem is in fact NP Complete, which means that it is equivalent to all other NP problems in the sense that a polynomial-time algorithm exists for converting it into any other one. Hence, if there was a polynomial-time algorithm for solving the constrained twenty questions problem, then a solution would be found for a host of famous problems, such as the Traveling Salesman and Chromatic Number problems. Consequently, it is quite unlikely that anyone will find such an algorithm for *every* instance of the constrained twenty questions problem. But this is unnecessary: as with many hard combinatorial optimization problems, it appears that there are suboptimal strategies which are nearly as good as the optimal one but immeasurably easier to find, and it is these strategies that we intend to analyze.

Before turning to other strategies, and more careful definitions, it is interesting to note that for relatively "small" problems, say for $M \leq 100$ and $N \leq 20$, one can in fact find the optimal strategy (and hitting time) using dynamic programming, and these empirical results (for a variety of setups $(p, \mathcal{X})$) support the contentions above and the more specific conjectures discussed below.

2.3.3 Strategies. Let ξ indicate the "true" hypothesis, so that ξ is a random variable with distribution p. Since ξ is the only source of randomness, we could take the probability space

as $\mathcal{M} = \{1, \ldots, M\}$, although we might also wish to consider the setup $\mathcal{X}$ itself as random. Consequently, we will assume a background probability space with probability measure P; thus $P(\xi = m) = p_m, m = 1, \ldots, M$.

Let X_n denote the outcome of the n'th test, $n = 1, \ldots, N$: $X_n = X_n(\xi) = X_{\xi,n}$. Given ξ, the distribution of $X_1, \ldots, X_N$ is degenerate. In particular, the X_n are conditionally independent given ξ. In a more general framework, we would want to consider families of "tests" which are conditionally nondegenerate, although perhaps still (conditionally) independent due to resulting simplifications in defining feasible strategies. In addition, we would want to consider tests X_n with more than two values, say values in $\{0, ..., L-1\}$, in which case the statements and conjectures to follow are simply modified by replacing $\log_2$ with $\log_L$.

Let us define a *strategy* as a function π from $\{0,1\}^N$ to the set of permutations of $\{1, \ldots, N\}$ such that π is *adaptive* in the sense that, for $x = (x_1, \ldots, x_N) \in \{0,1\}^N$, $\pi_1(x)$ is constant and $\pi_{n+1}(x) = \pi_{n+1}(x')$ whenever $x_i = x_i', i = 1, \ldots, n$. Equivalently, $\pi_1 =$ const., $\pi_2 = \pi_2(x_1), \pi_3 = \pi_3(x_1, x_2)$, etc. Let S_N denote the set of all strategies.

Notice that the number of strategies grows very fast with N. In fact, it is easy to see that

$$|S_N| = \prod_{n=0}^{N-2} (N-n)^{2^n}$$

Thus, e.g., $|S_4| = 576$ and certainly no empirical results can be obtained by searching among all strategies. However, as mentioned earlier, one can use dynamic programming to find the optimal strategy for N in the range $10 - 20$. This involves exploiting the enormous combinatorial reduction afforded by realizing that, for the applications in mind, M is very small compared to 2^N (in fact closer to order N), and hence the number of actual "histories" $x \in \{0,1\}^N$ that can be "seen" (i.e., generated by following a hypothesis with a given strategy) is approximately

$$\sum_{m=1}^{M} \binom{N}{X_{m,\cdot}} \leq M \binom{N}{N/2}$$

where $X_{m,\cdot} = X_{m1} + \cdots + X_{mN}$.

2.3.4 A Greedy Strategy and a Conjecture. We now focus on a particular strategy, which we call the *greedy strategy*; it is easy to compute and we conjecture it is nearly optimal in an appropriate sense. The idea is straightforward although the details are somewhat messy. We first choose the most informative test (about the true hypothesis) in the sense

of entropy; thus π_1 is that index k for which the entropy $H(X_k)$ of X_k is maximized, where

$$H(X_k) = -\sum_{x=0,1} P(X_k = x)\log_2 P(X_k = x)$$

(Notice that $P(X_k = x) = \sum_{m=1}^{M} p_m \delta_{\{X_{mk}=x\}}$ where δ_A is 1 if A is true and 0 otherwise.) Then choose $\pi_2(x)$ as the most informative test given $X_{\pi_1} = x, \quad x = 0, 1$, and so forth. Equivalently, π_1 is the argument of K which *minimizes* the expected value of entropy of the conditional distribution of ξ given X_k; $\pi_2(x)$ is the argument of k which minimizes the (conditional) expected value entropy of the conditional distribution of ξ given $(X_{\pi_1} = x, X_k), \quad x = 0, 1$; etc.

More specifically,

$$\pi_1 = \text{argmax}_{1\le k\le N} H(X_k)$$

and for $n \ge 2$, with $B(x_1, ..., x_{n-1}) = \{X_{\pi_1} = x_1, X_{\pi_2(x_1)} = x_2, \quad , X_{\pi_{n-1}(x_1,...,x_{n-2})} = x_{n-1}\}$,

$$\begin{aligned}\pi_n &= \pi_n(x_1, ..., x_{n-1}) \\ &= \text{argmax}_{1\le k\le N} H(P(X_k \in dx | B(x_1, ..., x_{n-1}))).\end{aligned}$$

(The assumption that $P(B) > 0$ is of no real concern since otherwise $B(x_1, ..., x_l)$ determines a *single* hypothesis for some $l < n-1$, after which no other tests are informative (i.e., all the conditional distributions are degenerate) and the remaining $\pi_n, n \ge l$, may be arbitrarily defined for sequences beginning with $(x_1, ..., x_l)$.)

An equivalent definition of this strategy can be given in terms of the conditional distributions of ξ given the previous outcomes. Let $H(P(d\xi|X_k))$ denote the entropy of the conditional distribution of ξ given X_k. Thus for $x = 0, 1$:

$$H(P(d\xi|X_k = x)) = -\sum_{m=1}^{M} P(\xi = m|X_k = x)\log_2 P(\xi = m|X_k = x)$$

where

$$P(\xi = m|X_k = x) = (p_m \delta_{\{X_{mk}=x\}}) / \sum_{j=0}^{M} p_j \delta_{\{X_{jk}=x\}}.$$

Then

$$\begin{aligned}\pi_1 &= \text{argmin}_{1\le k\le N}\ E(H(P(d\xi|X_k))) \\ &= \text{argmin}_{1\le k\le N} \sum_{x=0,1} H(P(d\xi|X_k = x))P(X_k = x)\end{aligned}$$

and for $n \ge 2$

$$\pi_n(x_1, ..., x_{n-1}) = \text{argmin}_k\ E(H(P(d\xi|B(x_1, ..., x_{n-1}), X_k)))$$

The conditional distribution is

$$P(\xi = m|B, X_k = x) = \frac{p_m I_{mk}(x_1, ..., x_{n-1}, x)}{\sum_{j=1}^{M} p_j I_{jk}(x_1, ..., x_{n-1}, x)}$$

where $I_{jk} = 1$ if $X_k = x$ and $X_{j\pi_i}(x_1, ..., x_{i-1}) = x_i$ for each $1 \leq i \leq n-1$, and $I_{jk} = 0$ otherwise. The fact that these definitions are equivalent follows from the identity below in which $B = B(x_1, ..., x_{n-1})$ and $1 \leq k \leq N$:

$$H(P(d\xi|B)) = H(P(X_k \in dx|B)) + E(H(P(d\xi|B, X_k))|B).$$

If the strategy were executed *on line*, the computational requirements for choosing test π_n (having chosen $\pi_i, i \leq n-1$) would be modest: given we have performed n tests the choice of the next test only depends on the corresponding *current* distribution $p^{(n-1)} = P(d\xi|X_{\pi_1}, ..., X_{\pi_{n-1}})$ on hypotheses. In fact we need only know the *support* $A \subset \{1, ..., M\}$ of $p^{(n)}$ to determine its values:

$$p_m^{(n)} = p_m / \sum_{j \in A} p_j.$$

Now since the X_n are just binary variables, and since the entropy of a two-point distribution $\{\alpha, 1-\alpha\}$ is monotonically increasing as $|\alpha - .5|$ decreases, we see that

$$\pi_n = \text{argmin}_k \, |.5 - \sum_{m=1}^{M} p_m^{(n-1)} \delta_{\{X_{mk}=1\}}|$$

Thus we choose the test which most nearly divides the "active" hypotheses into two groups of equal probability, where m is an active hypothesis at stage n if $p_m^{(n-1)} > 0$. Actually, the strategy is generally not unique, and π_n may be chosen arbitrarily in the case of multiple minima.

Notice that the greedy strategy is a "first order" strategy in that the choice of π_n depends only on the *individual* (conditional) distributions of the random variables $X_1, ..., X_N$. In particular, it does not depend on the *joint* distribution:

$$P(X_1 = x_1, ..., X_N = x_N|X_{\pi_1}, ..., X_{\pi_{n-1}}) = \sum_m \prod_{k=1}^{N} \delta_{\{X_{mk}=x_k\}} p_m^{(n-1)}$$

It would be interesting to show that the greedy strategy is at least (nearly) optimal among all first order strategies, and perhaps to explore higher order strategies, such as those involving pairwise test interactions (say based on the resulting entropy after asking two questions).

Let τ_g, τ_{opt} denote the decision times for the greedy and optimal strategies respectively. We believe that τ_g performs very well compared to τ_{opt}. In fact, we make the following—

CONJECTURE: *$\Delta(p,\mathcal{X}) \equiv E(\tau_g) - E(\tau_{opt})$ is "small" compared to $E(\tau_{opt})$ and perhaps uniformly small over all $(p,\mathcal{X})$.*

This contention is based partially on the aforementioned experimental evidence, in which we compared the greedy and optimal hitting times for various choices of $p,\mathcal{X}$, and partly on the result below. If we can establish this result, perhaps even a probabilistic version involving a doubly stochastic system with an (independent) distribution over matrices $\mathcal{X}$, then this would help to provide the theoretical and practical foundation for the rigid body recognition problem, in which case the tests would correspond to probes and the hypotheses to both object-aspect pairings and perhaps entities corresponding to clutter and background noise; see §2.3.7.

Theorem. *If all tests are available (i.e., $M = 2^N$), then*

$$E(\tau_g) \leq H(p) + 1.$$

In particular, in view of the Huffman coding result, this implies that the mean number of tests for the greedy strategy is within at most one unit of that for the optimal strategy.

Proof. First, it is fairly straightforward to prove that

$$P(\tau_g > n | \xi = m) \leq \delta_{\{p_m < 2^{-n}\}}$$

for every n,m. For example, if some p_j is at least $1/2$, then clearly the corresponding hypothesis is identified in exactly one step, and this reasoning can be extended from $n = 1$ to any n by exploiting the fact that all "splits" are available at every stage. Assuming this,

$$\begin{aligned}
E(\tau_g) &= \sum_{n=0}^{\infty} P(\tau_g > n) \\
&= \sum_{n=0}^{\infty} \sum_{m=1}^{M} P(\tau_g > n | \xi = m) p_m \\
&\leq \sum_{n=0}^{\infty} \sum_{m=1}^{M} p_m \delta_{\{p_m < 2^{-n}\}} \\
&= \sum_{m=1}^{M} p_m \sum_{n=0}^{\infty} \delta_{\{p_m < 2^{-n}\}} \\
&= \sum_{m=1}^{M} p_m \sum_{n=0}^{[-\log_2 p_m]} 1 \\
&= \sum_{m=1}^{M} p_m (1 + [-\log_2 p_m]) \\
&\leq H(p) + 1.
\end{aligned}$$

where [ ] stands for the greatest integer function.

2.3.5 The Markov Chain of Active Sets. From here on we shall restrict our attention only to *regenerative strategies* such as the greedy one, i.e., those for which π_n only depends on the current distribution over hypotheses, or, what is the same, the set of active hypotheses. Clearly, the optimal strategy must also be of this type. We may regard these strategies as determined by a function ν which returns a coordinate value $\nu(p) \in \{1,...,N\}$ for each probability vector p on $\mathcal{M}$. More conveniently, since $p^{(0)}$ is given, we can take ν as a mapping $\nu : \mathcal{A} \to \{1,...,N\}$, where $\mathcal{A}$ denotes the set of subsets of $\mathcal{M}$.

Given a strategy π, let Z_n denote the set of active hypotheses after n tests:

$$Z_n = \{m : P(\xi = m | X_{\pi_1}, ..., X_{\pi_n}) > 0\}$$

Then $Z_0 = \{1,...,M\} \supset Z_1 \supset Z_2 \cdots$, and $\{Z_n,\ n = 0,...,N\}$ is a Markov chain with state space $\mathcal{A}$. The transition function is

$$P_\pi(Z_{n+1}|Z_0,...,Z_n) = \frac{\sum_{m \in Z_n} p_m \delta_{\{X_{\nu(Z_n)} = x\}}}{\sum_{m \in Z_n} p_m}, \qquad \text{with}$$
$$Z_{n+1} = Z_n \cap \{m | X_{m\nu(Z_n)} = x\}, \qquad x \in \{0,1\}.$$

All singleton subsets are absorbing states and the decision time τ is the entry time to this set, i.e. $\tau = \inf\{n \geq 0 | |Z_n| = 1\}$.

2.3.6 Other Cost Functionals. So far we have measured performance of a strategy in terms of the expected time to a decision, a natural and compelling criterion. However, there may be other cost criteria which have either practical or mathematical advantages. In particular, we have explored another type of cost functional, which, in some cases, reflects the actual *computational* cost of the search itself, and might be more analytically tractable than the hitting time. Let Z_n be as above, except set $Z_n = \emptyset$ for $n > \tau$. Let τ_m denote the extinction time for hypothesis m:

$$\tau_m = \inf\{1 \leq n \leq N+1 : m \notin Z_n\}$$

Then $\max_m \tau_m = \tau + 1$ and $\tau_m > n$ if and only if $m \in Z_n$. It follows easily that

$$\sum_{n=0}^{N} |Z_n| = \sum_{m=1}^{M} \tau_m$$

Now after each test is evaluated, it is usually necessary to "prune" the list of active hypotheses, and we might consider a performance criterion based on the assumption that

this is the major computational cost in executing an algorithm such as the greedy one, or that all other costs (evaluating tests, updating distributions, etc.) are more or less common to all other strategies of interest. Consequently, one might consider the optimal strategy to be that which minimizes

$$E(C), \qquad C = \sum_{n=0}^{N} |Z_n|$$

This is the expected total number of entries of the matrix $\mathcal{X}$ which are visited for checking consistency with test results; equivalently, it is the sum of the extinction times. Thus, e.g., in the equally likely case, it is easy to show that

$$E(C) = M^{-1} \sum_{n=0}^{N} \sum_{x \in \{0,1\}^n} |Z_n(x)|^2$$

where $Z_n(x)$ denotes the set of active hypotheses given $(X_{\pi_1}, ..., X_{\pi_n}) = x$.

2.3.7 Back to Object Recognition. Consider again the situation described in §2.1 and §2.2: we are given a suitable mathematical characterization of a set of rigid objects together with an image containing one or more of these objects in any spatial orientation; we wish to identify which objects appear in the image, especially in the presence of noise, clutter, and other sources of degradation.

2.3.7.1 Clutter and Background Models. Throughout this section we shall suppose that a field of view subimage is fixed, somewhat larger than the size of the largest object-aspect silhouette, and our goal is to determine whether any of the objects is "centered" within the field of view with respect to some reference point. Or perhaps we wish to determine whether any of the objects appears anywhere (entirely) within the field of view. Let us also assume that the set of rotations and translations ("offsets") has been suitably discretized, and hence there are a finite number of "hypotheses" corresponding to the distinct object-aspect pairings. We may also wish to have hypotheses corresponding to clutter other than that induced by the offsetted objects themselves, and certainly one or more hypotheses corresponding to the event that no object is present at the reference point. Consequently, we might consider a slightly different situation from the one with M individual hypotheses, and consider the set $\mathcal{M}$ of hypotheses as decomposed into a disjoint union of sets $\mathcal{M}_k$, where for example $\mathcal{M}_1$ denotes a composite hypothesis consisting of realizations of background and clutter, and each $\mathcal{M}_k, k \geq 2$, is the (possibly) composite hypothesis consisting of presentations of object k, perhaps only "at" the reference point or perhaps together with offsets, in which case these offset hypotheses would not appear in $\mathcal{M}_1$. In this context,

assuming that we are not interested in separating one presentation of a given object from another, we would then take

$$\tau = \inf\{n \geq 1 : Z_n \subseteq \mathcal{M}_k, \text{some } k\}.$$

2.3.7.2 Probes. The tests $\{X_n\}$ we have in mind are the types of "probes" we described earlier. Let $\{Y_t, t \in T\}$ denote the (ideal) intensity or depth values for a field of view T, which may or may not contain an object aspect pairing. For *binary* templates (say $Y_t = 0$ outside silhouettes and $Y_t = 1$ inside silhouettes), typical tests would correspond to individual pixels, such as $X = \delta_{\{Y_t=1\}}$, $t \in T$ (figure/ground dichotomy) and to pairs of pixels, such as $X = \delta_{\{Y_t \neq Y_s\}}$, $t, s \in T$ (transition/no transition). (Recall that for the actual grey level images it would be necessary to set a threshold C to declare that "$Y_t = Y_s$" when $|Y_t - Y_s| < C$.)

For grey level templates, such as with range data corresponding to (ideal) depth values, we might choose *relational* tests based on pairs of pixels t, s of the form $X = 0, 1, 2, 3$ respectively if (t, s) is an (inside, inside), (outside, inside), (inside, outside), or (outside, outside) pair of image locations. Similarly, tests based on four locations might be of the form $X = \delta_{\{|Y_t - Y_s| < |Y_u - Y_v|\}}$, the interpretation for range data being that s, t represent spatial points which are "closer together" than those represented by u, v.

2.3.7.3 Decision Making with Noise: Floating Thresholds. An important issue is data conversion, transforming the observed intensity values, often degraded and assuming many values, into elementary test values for comparison with the (ideal) stored values. We have experimented with "floating thresholds," and propose to supply a rigorous statistical justification for this approach.

Basically, the idea (in the binary case) is the following. Given that a certain hypothesis is "active" at a given image location, we must decide whether or not to keep it active after collecting information in the form of test results. Due to the presence of noise and other degradation factors, we do not anticipate updating the posterior distribution over hypotheses after *each* individual test, but rather after a series of tests, which, as in §2.2, may be regarded as a node on a decision tree. For example, we might order the entire collection of tests in accordance with some strategy and then group these tests into batches. Now given we are considering a particular hypothesis in a field of view, and given the raw test data, we order the outcomes and select a threshold for which the number of tests which are then assigned the label "1" matches the number of "1"s in the stored (ideal) test sequence

values for the particular hypothesis. Thus the pending hypothesis is given the "benefit of the doubt" and at low noise levels the observed and stored strings will necessarily coincide. We conjecture that the probability of detection error is then minimized under an appropriate model for the image formation process.

3. RECOGNITION OF NONRIGID OBJECTS

3.1 Introduction.

Our approach to the recognition of rigid objects makes essential use of a presumed fixed and *a priori* known structure, both for problem-specific representations and for search strategies. The approach is workable when the presentation of objects is highly constrained; we have so far explored examples with only one (rotation) or two (rotation and range) degrees of freedom. Of course, there are already many practical problems in which this set up is realistic, and it is undoubtedly possible to make extensions (perhaps involving *ad hoc* estimates of scale and orientation) to several more degrees of freedom. However, many recognition tasks involve shapes with essentially infinite degrees of freedom, such as the presentation of hands or hearts or coronary arteries in medical pictures, or the presentation of numerals in handwritten zip codes, to name just a few examples. For these problems we will require entirely different representations and search strategies.

The basic idea behind our approach to nonrigid object recognition can be viewed in either of two ways, and we shall refer to both points of view in the following discussion. From the more intuitive point of view, our object models are *templates* together with a *distribution* on *deformations* of these templates. The templates are prototypical examples of the object class. The ensemble of presentations of a particular object is thereby modeled as the result of acting on a template with a random deformation. The advantage of this approach is that it conveniently captures both the *global regularities* (embodied in the template) and the *typically local departures from the prototype* that characterize a particular instance of an object. It is important that the deformations are mostly local, because we can then describe them by a (local) random field, and this has essential computational advantages.

The idea of deformable template models for nonrigid objects is certainly not new. Many authors have taken the same basic approach (see for example Bajcsy and Kovacic [6], Burr [8], Fischler and Elschlager [15], and Widrow [49]), although there is considerable variation in the details, and, in general, the use of random fields to describe the deformation seems to be new. We began the study of our approach with some examples of biological shapes (Amit, Grenander, and Piccioni [3], Grenander, Chow, and Keenan [30], and Knoerr

[37]), and have recently begun exploring several other examples, as will be discussed below.

The other point of view referred to above is that of hidden Markov models. The application of a random deformation to a template produces a random structure. If the deformation mechanism is a local random field, then so is the resulting structure. Mostly, we use *homogeneous* deformation fields, and leave it to the template to capture global and object-specific structure. The resulting *random* structure is inhomogeneous; the structure models are then inhomogeneous (nonstationary) Markov random fields. In the one dimensional examples below of coronary arteries and handwritten numerals, these are second order $\mathbb{R}^2$−valued Markov processes, with branching graph structures (see §3.2.). The actual observations are manifestations of these structures in a grey-level image, and thus the overall observation model is a hidden Markov model. Our approach, then, can be viewed as an adaptation of the highly successful hidden Markov approach to speech recognition (Bahl, Jelinek, and Mercer [5], Lee [39], and Rabiner [45]). For application to vision, we extend the Markov models from conventional one dimensional models to random fields with branched linear graphs, and with two and three dimensional graphs.

3.2 One dimensional structures.

3.2.1 Examples. We have studied two prototypic applications. One is the identification of coronary arteries in angiograms. The other is the identification of handwritten numerals. The angiogram project is in collaboration with Dr. Jonathan Elion who is a cardiologist at Brown University Medical School and is an expert in the applications of digital image processing techniques in medical images. The goal is to locate and identify instances of the major coronary vessels in coronary arteriograms. These arteries are highlighted by injection of contrast dye in their proximal portions, and thereby show up as relatively dark regions in (digitized) X-ray images (see Figure 3.1). The motivation is to support increasingly automated analysis of medical images. More specifically:

1. Several approaches to the computer based assessment of coronary artery structure and physiology from radiographic images have recently been developed. Among these are quantitative measures of coronary stenosis, three-dimensional reconstruction of the coronary tree, assessment of regional contractility, and measurement of coronary blood flow. Widespread application of these approaches has not been realized because of the lack of a reliable means of automatic computer identification of key anatomic landmarks. The current state of the art for most computer programs that analyze cardiac images requires an expert operator to interact with the program and identify the landmarks. This introduces opportunities for inter- and intra-observer

variability, and limits the utility of the analysis during the actual performance of the cardiac catheterization. Automated identification of coronary vessels would provide the needed landmarks for many of these applications.

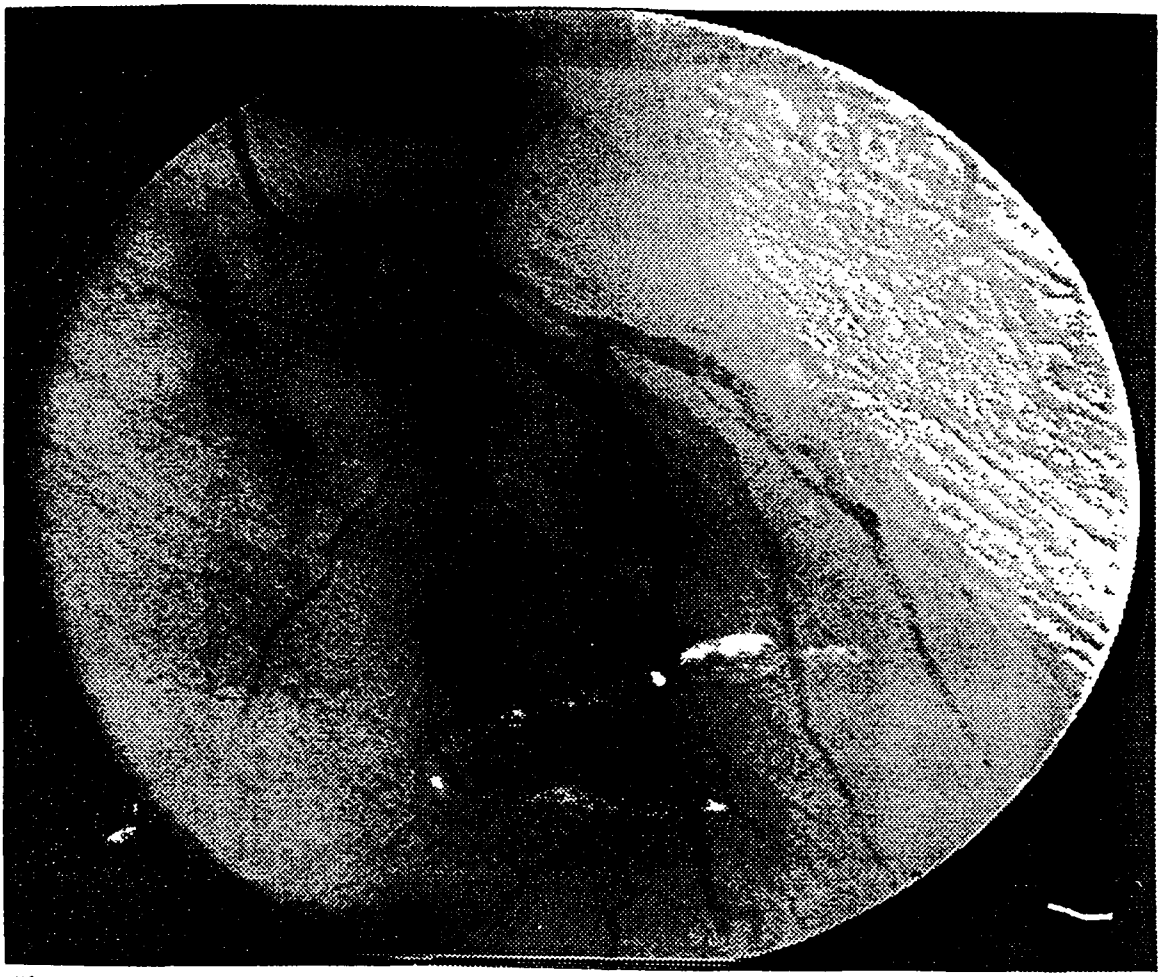


Figure 3.1. Angiogram showing right anterior oblique (RAO) coronary artery

2. Arterial lesions are best viewed and most accurately assessed when the X-ray beam is perpendicular to the artery. Modern angiogram equipment is capable of imaging in arbitrary planes, and can be moved during an imaging session. This facility is generally not used because it is difficult for the clinician to ascertain the appropriate orientation. This process could be automated if accurate machine identification of the coronary vessels could be achieved.

3. Modern digital X-ray techniques produce very large volumes of image data from each diagnostic session. It is not feasible for the clinician to review coronary vessel status in every collected frame. Automated search, and ultimately, automated highlighting

of lesions (generally seen as a narrowing of the artery) would be a valuable clinical adjunct.

There are, as well, a host of important applications to handwritten numeral recognition. These include automatic zip code reading, reading and verification of dollar amounts (so-called courtesy numbers) on hand-drawn checks, and verification of dollar amounts on credit card receipts. More generally, the problem is prototypical of finding one-dimensional structures in images, including surface and shading discontinuities, and object boundaries. Figure 3.2 shows a set of courtesy numbers and illustrates the variation in size, orientation, and shape that must be accommodated.

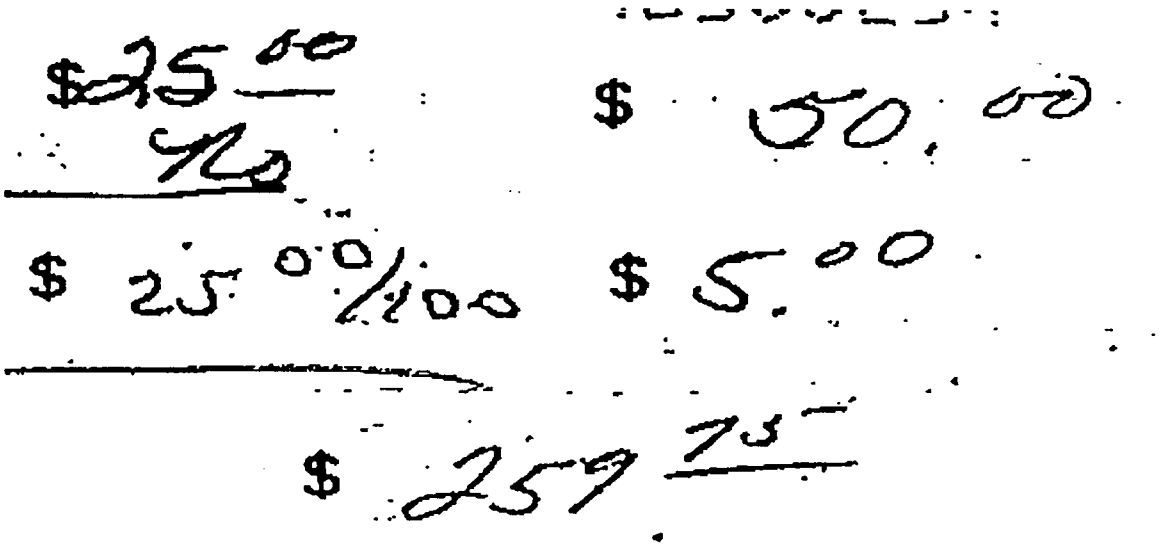


Figure 3.2. Courtesy numbers from hand-drawn checks (thresholded images)

3.2.2 Shape models. Global structure is coded via certain canonical examples that serve as templates. The object ensemble is modeled as the result of applying a random deformation to the template(s). With an eye towards the recognition task, and the associated computational algorithms, we approximate the templates by linear splines; see Figures 3.3 and 3.4 for example artery and numeral templates together with their spline approximations. For now, we have simply chosen the knots (designated x_k^T, $k = 1, 2, ...$; T for 'template') by hand, via an interactive mouse-driven program. As will be obvious from the ensuing discussion it is important to include the natural "critical points", such as discontinuities in derivatives as well as branch points and junctions. There is a natural continuum formulation which is illustrative, and in fact helpful in guiding the scaling of certain of the distribution parameters; this will be discussed shortly.

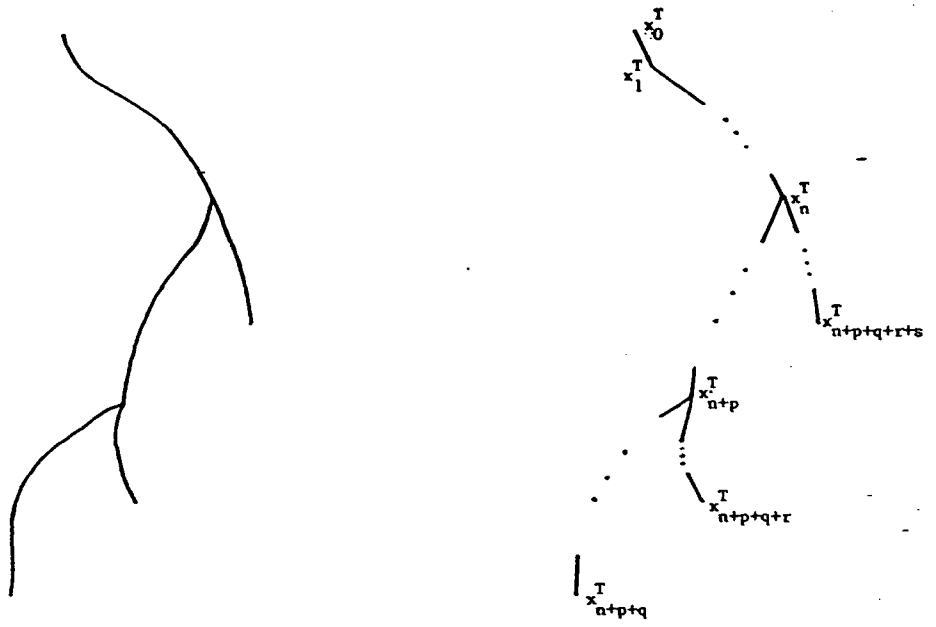


Figure 3.3. Artery template and its spline representation

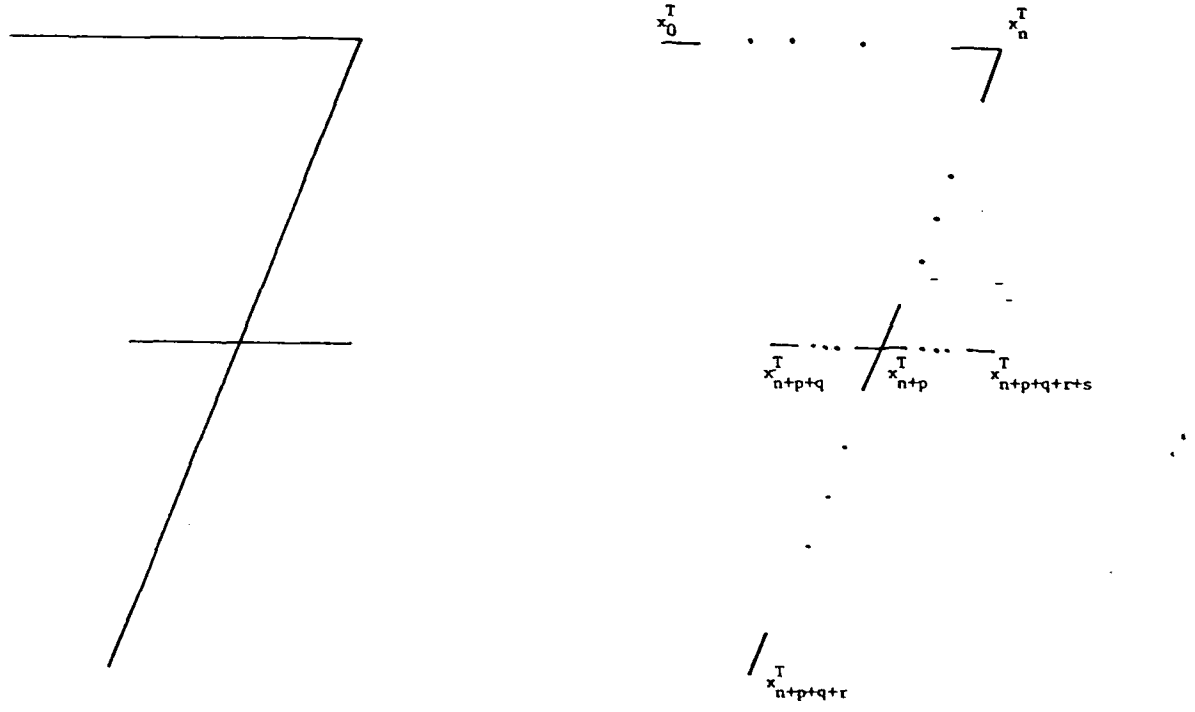


Figure 3.4. Numeral template and its spline representation

There may be only one template for a class, but we anticipate more typically several templates. (Consider, for example, the European three-stroke seven versus the American two-stroke seven.) In any case, the ensemble of objects in a class is generated by applying random deformations to the templates. These deformations are modeled as nearest neighbor group-valued Markov processes, with one site corresponding to each segment of the spline-approximated template. We will denote by S_k a random affine transformation to be applied to the k^{th} segment, which runs between x_{k-1}^T and x_k^T. We model the collection of transformations, $S_0, S_1, ...$, as a nearest neighbor ("first order") Markov process with graph that is the dual of the template graph: i.e. two transformations S_a and S_b are neighbors if the corresponding line segments share an end point. Finally, the joint distribution on $S_0, S_1, ...$ is constrained to preserve the continuity of the template; connected line segments are mapped to connected line segments.

For these one-dimensional models it is more convenient to take the hidden Markov model point of view; that is, it is easier to work directly with the induced distribution on the knots, $x_0, x_1, ...$. (The absence of the superscript T will distinguish these *random variables* from the template knot locations $x_0^T, x_1^T, ...$.) It is easy to check that the induced distribution is *second order Markov:* the placement of x_3, for example, will depend on both x_1 and x_2, since these together carry information about the transformation S_2, which in turn determines the distribution on S_3 and hence on x_3. Thus we arrive at a second order Markov process taking values in R^2. The associated dependency graph for the artery example of Figure 3.3 is shown in Figure 3.5. It is important to realize that, whereas the distribution on transformations may be chosen to be homogeneous, the resulting distribution on knots is inhomogeneous. In particular, the global template structure enters in when the transformations are actually applied to the template line segments. This is then reflected in the distribution on knots.

In our preliminary experiments we have used, exclusively, *Gaussian* distributions on the knots $x_0, x_1, ...$. To build in rotation and scale invariance we impose *uniform* and independent distributions on x_0 and x_1, and then, conditioned on these:

$$x_2 = x_1 + A_1(x_1 - x_0) + \eta_1 \tag{3.1}$$

where A_1 is a fixed (*template specific*) 2×2 matrix, and η_1 is symmetric Gaussian in R^2. Similarly, $x_3 = x_2 + A_2(x_2 - x_1) + \eta_2$, and so on. (Strictly speaking, at the first knot point or branch point there is a dependence not only on the branch point and its immediate predecessor, but also on any other nearest neighbors of the branch point. Refer, for example, to Figure 3.5. We have simplified somewhat by ignoring these latter dependencies. This gives us a one-sided "causal" representation of the random structure.)

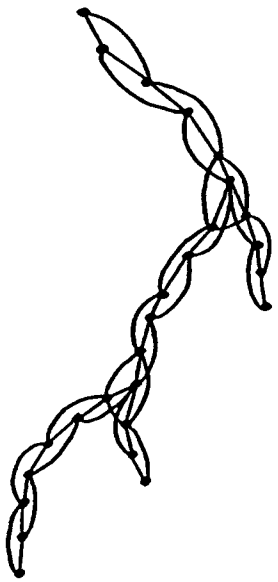


Figure 3.5. Dependency graph for artery shape model

As we have indicated earlier, the knots are chosen rather arbitrarily, and by hand. In particular, the resulting segment lengths are not equal. We typically put more knots in regions of high curvature, for example. It is natural, and important, that the variance of a knot, say that of x_2, depend on the length of the associated template segment, x_1^T to x_2^T. Referring to equation (3.1), this amounts to placing an inhomogeneous variance on η_k, say $\sigma_k^2 I$, where I is the 2×2 identity. How should σ_k scale with the template segment length $|x_{k-1}^T - x_k^T|$? The answer can be found by taking the continuum limit of the discretization procedure, forcing the length of the longest segment to zero. The result is a *random* and *branching* continuously differentiable curve, whose derivative is an inhomogeneous diffusion. In carrying out this procedure one discovers the correct scaling of the standard error: $\sigma_k \propto |x_{k-1}^T - x_k^T|^{3/2}$.

3.2.3 Data models. The modeling task is completed by specifying a distribution on observable grey levels conditioned on a particular instance of the object class. The product of this distribution with the shape distribution for the object class, developed above, is then the joint distribution on shapes and their observed presentations. It is obvious from Figure 3.1 that the width and contrast of an artery is highly variable. This is certainly the rule in challenging recognition tasks: the actual grey-level presentation of an object is not

well accounted for by any single parametric distribution, even when the shape, range, and orientation are fixed. (See related discussion in §2 above.)

For both arteries and handwritten numerals, a minimal characterization of the grey-level presentation specifies that it is (locally) darker than the background area in its immediate vicinity. Along a segment, x_{k-1} to x_k, we expect to find intensities that are typically darker than those found along parallel segments to either side; see Figure 3.6. Using y_i, z_i, and w_i for the intensities encountered in the three rectangular regions illustrated in Figure 3.6, we model the grey-level likelihood, conditioned on the locations x_{k-1} & x_k, as

$$p^{\alpha}(1-p)^{N-\alpha}$$

where

$$\alpha = \sum_{i=1}^{N} 1_{\{y_i < z_i \& y_i < w_i\}}.$$

α is the number of times y_i is darker than both z_i and w_i, and p is an *a priori* estimate of the probability of this event. p is a rough characterization of the signal to noise ratio.

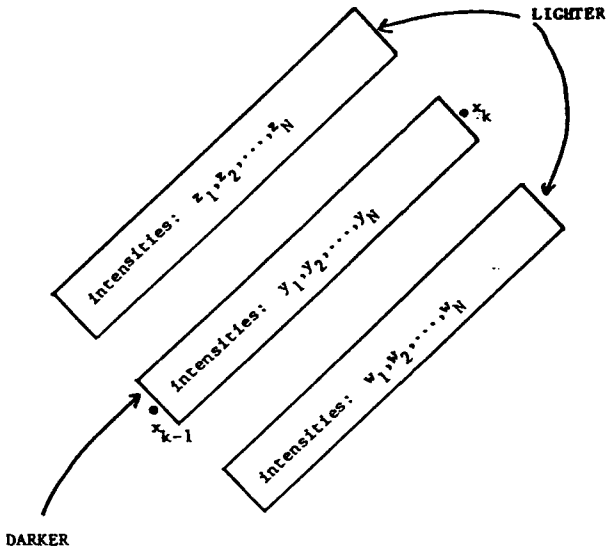


Figure 3.6. Nonparametric characterization of grey-level presentation of artery or numeral segment

We have found this likelihood to be computationally efficient and effective in handling rather drastic contrast changes such as those seen in Figure 3.1. But many variations are possible, and better data models will undoubtedly emerge. (In fact, very detailed models of the grey-level presentation of the object are possible, within this same Markov framework; see the work of Cooper et. al. [10]. on boundary detection.)

3.2.4 Matching algorithm. We wish to find all instances of each object type in a given grey-level picture. As an example, consider the two templates shown in the left panel of Figure 3.7. These were placed in an artificial "noisy" scene and the composite was distorted by folding before imaging (right panel, Figure 3.7). The problem is to find all instances of object types "1" and "2".

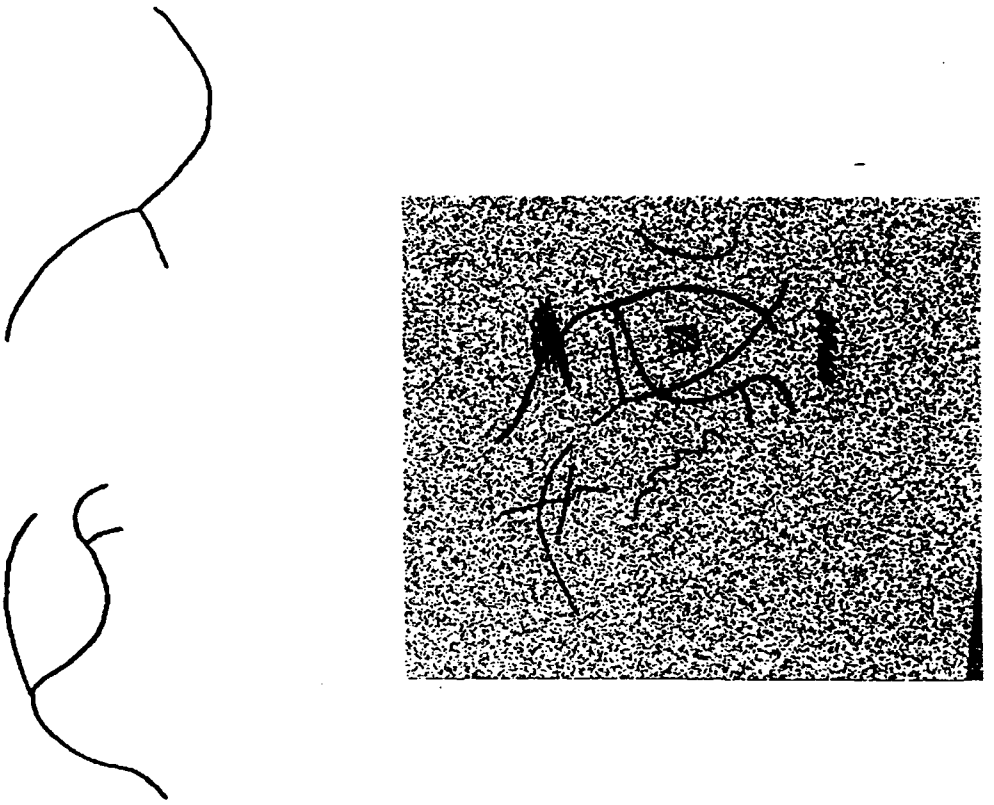


Figure 3.7. Left panel: two hand drawn templates. Right panel: Placement of templates in artificial scene. Templates were distorted by slightly folding the scene before imaging.

Suppose that we screen (*liberally*) the entire image for possible end points of line segments. We could then attempt to "track", from each of these purported end points, each of the object types ("1" and "2" in the example). More specifically, we could attempt to maximize the likelihood of knot placements under each object model, conditioned on the fixed end point and on the grey-level data. Careful examination of the above shape models indicates that this maximum can be computed via *dynamic programming*. The end result is the *most likely* match of a given template to the grey-level image, under the initial constraint on the end point. This match has an associated likelihood which serves as a measure of the strength of the hypothesis that the object is indeed present with the intended end point. This was carried out in the synthesized scene shown in the right panel of Figure 3.7. The result was a collection of likelihoods that separated into the two correct matches (high relative likelihoods) versus lower likelihoods of incorrect matches; see Figure 3.8.

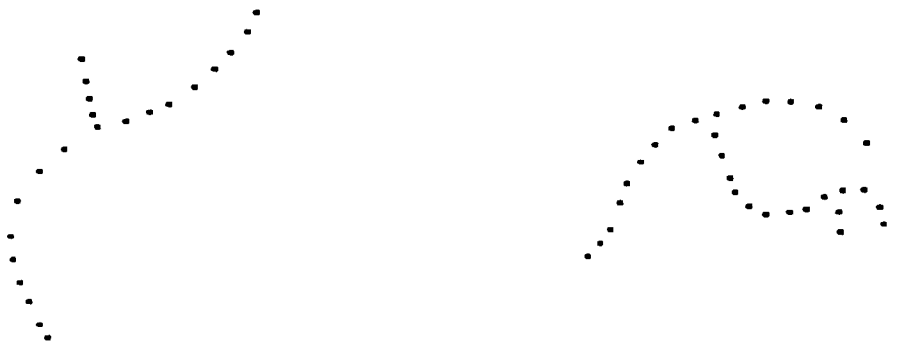


Figure 3.8. High likelihood matches for Figure 3.7 scene

The computational complexity of the dynamic programming algorithm is a function of the order of the underlying Markov process that serves as our shape model. In particular, the number of operations is:

$$M \cdot \Sigma^{O+1}$$

where M is the number of knots, Σ is the size of the state space of the knots, and O is the order of the model, order two for the artery and numeral models introduced here. In principle, Σ is the number of pixels in the image, but under the Gaussian model, all but a few pixels are essentially impossible, given the placements of the two previous knots. At the inner most loop in the dynamic programming algorithm, values of the two preceding knots

are fixed and hence most of the Σ pixel locations can be ignored. This and other similar *pruning* assumptions make the computations feasible.

3.2.5 Coarse-to-fine search. Figures 3.7 and 3.8 illustrate the ability of the algorithm to ignore confounding noise and to "tunnel" through gaps. However, the necessity of pruning makes the algorithm vulnerable to such effects, and we often find suboptimal matches in experiments with real images. Coarse-to-fine search strategies can substantially constrain the reasonable matches. The idea is to condition on the location of two or more possible end points of a hypothesized object. Candidate end points are relatively easy to find. These can then be taken in *pairs*, and two end points of a template can then be constrained to match these locations. The resulting *conditional distribution* on the remaining knots is then used for the dynamic programming search. Preliminary experiments have been encouraging; see Figure 3.9.

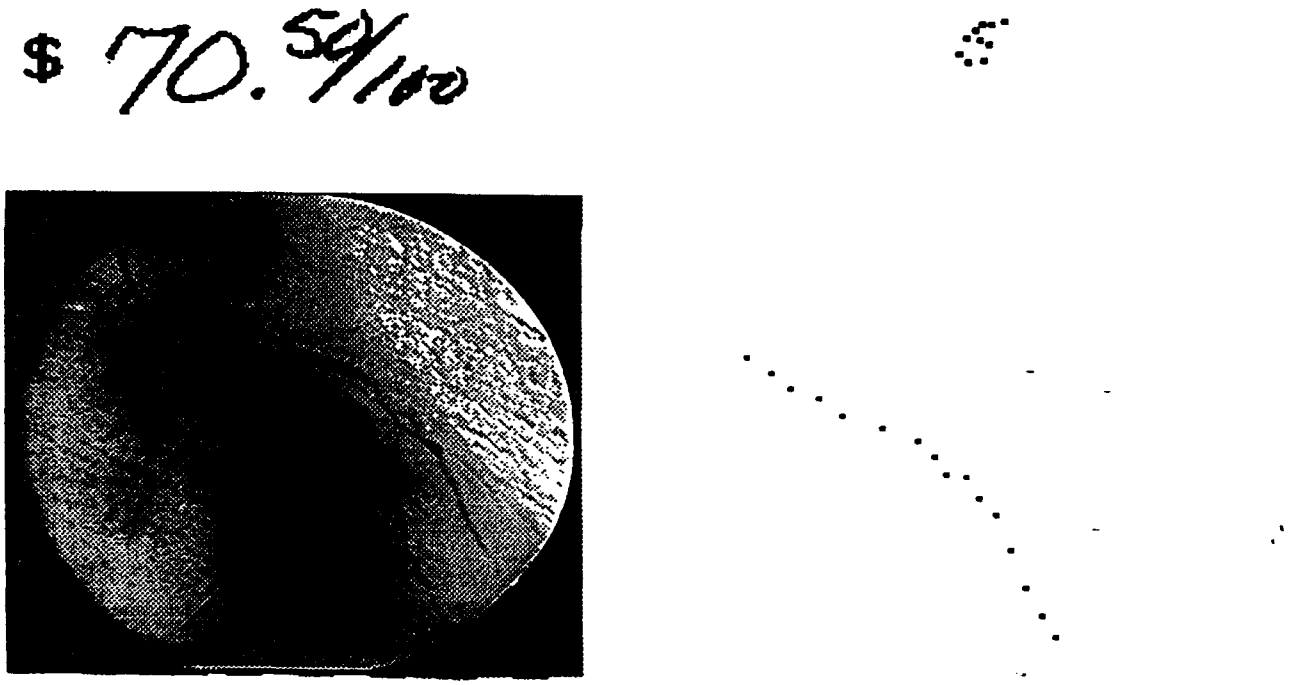


Figure 3.9. Matches conditioned on two endpoints

This same reasoning can be applied at several resolutions. The lowest resolution is on pairs, or perhaps triples, of end points. Conditional on these, other critical points are sought. The procedure continues to the highest resolution which amounts to tracking curve segments under multiple positioning constraints. Very much *unlike* the situation in two

and three dimensions, where distributions at course resolutions must be *approximated*, the one-dimensional structures of the models proposed here admit *exact* representations at all resolutions.

3.2.6 Model extensions. It will most likely be necessary to broaden the class of structure models. For example, by beginning with *homogeneous* random fields for the description of deformations we preclude the possibility of certain degrees of freedom having naturally higher variance than others. But it is clear, for example, that the angles at junctions of strokes in handwritten characters are more variable than angles between spline segments internal to a given stroke. The above framework needs to be extended to better accommodate the natural degrees of freedom in the object class. This will likely require that we introduce non-Gaussian distributions, and will thereby complicate the calculation of coarse resolution distributions (see above discussion of coarse-to-fine search).

We anticipate that it will still be possible to calculate closed form expressions for coarse level distributions, although this may now require the use of automated symbolic manipulation, as is available with Macyma, Mathematica, and other similar software packages. For a given model, these are "off-line" calculations that need only be carried out once.

3.2.7 How rich is the class of hidden Markov models? A common ingredient in our work on image processing and image analysis has been the use of Markov processes, of the usual one-dimensional variety as well as more general branched-type processes, such as those introduced above, and two and three dimensional random fields. These processes have been used to model structure and variability of image attributes and objects. Thus we have used two-dimensional Markov random fields as models for grey-level distributions and boundary placement ([18,29,25]), for modeling textures ([20,22]), and for modeling isotope concentration maps in single photon emission computed tomography ([24,40,43]), and more recently, one, two and three dimensional random fields for modeling shapes in reconstruction and recognition experiments ([3,30,37]). In these applications, as well as in the application of Markov models to speech recognition, what are actually observed are *functions* (sometimes random) of the Markov process. The observables are thereby "hidden Markov". The resulting observation process itself is typically not Markov; in fact, it is likely to have a very complex dependency structure. Indeed, this is what is behind the utility of hidden Markov models. The model is built from *local pieces* (a local Markov process—or random field—and a local observation equation), but can account for highly complex and nonlocal structures in the observations. The question arises as to how general is the class

of hidden Markov models. This and related questions have been explored in some recent work with Hans Kunsch and Athanasios Kehagias.

To be precise, consider, for example, the class of first order hidden Markov models in one dimension. Let $\{x_t\}_{t=1}^{\infty}$ be a stationary first order Markov process with *finite* state space Ω_x. Let $f : \Omega_x \to \Omega_y$ where Ω_y is also finite, and typically $|\Omega_y| < |\Omega_x|$. Then $y_t \doteq f(x_t)$ is a hidden Markov model. How rich is the class of such processes? Apparently very rich, as indicated by the following relatively easy result:

Theorem. *Let $\{y_t\}_{t=-\infty}^{\infty}$ be a stationary process with finite state space Ω_y. There exists a sequence of finite-state stationary hidden Markov models $\{x_t^N\}_{t=-\infty}^{\infty}$, Ω_x^N, $f^N : \Omega_x^N \to \Omega_y$, such that the processes*

$$y_t^N \doteq f^N(x_t^N)$$

converge weakly to $\{y_t\}_{t=-\infty}^{\infty}$.

In fact, the multi-dimensional version of this is also true: We can approximate any stationary process on L^d (the d-dimensional square lattice) by a sequence of *nearest-neighbor* hidden Markov random fields. By nearest neighbor, we mean a pair-clique system of nearest horizontal and vertical neighbors only. Thus:

Theorem. *Let $\{y_t\}_{t\in L^d}$ be a stationary process with finite state space Ω_y. There exists a sequence of finite-state stationary nearest-neighbor hidden Markov random fields $\{x_t^N\}_{t\in L^d}$, Ω_x^N, $f^N : \Omega_x^N \to \Omega_y$, such that the processes*

$$y_t^N \doteq f^N(x_t^N)$$

converge weakly to $\{y_t\}_{t\in L^d}$.

Part of the motivation is to establish the richness of this class of models for practical applications. Evidently, very little is lost in restricting oneself to hidden Markov models. In addition, there is the intriguing possibility of rendering textures for computer graphics and simulations, and of modeling one-dimensional signals such as speech waveforms. For these modeling purposes, we need a corresponding estimation theory to guide the selection of an appropriate hidden Markov model given observations of an arbitrary stationary process. Consider again the one-dimensional problem, and suppose that $\{y_t\}_{t=-\infty}^{\infty}$ is ergodic with state space $\Omega_y = \{1, 2, ...q\}$. Let M_N be the class of all strictly positive $N \times N$ transition probability matrices on $\Omega_x^N \doteq \{1, 2, ...N\}$, and let $f^N(x) = f(x) = x \bmod q$. If $\{x_t^N\}$ is the stationary Markov process associated with an element of M_N, then $y_t^N = f(x_t^N)$ is hidden

Markov with state space Ω_y. Given an observed sequence $y_1, y_2, ...y_n$ of the $y-$process, we seek to "fit the data" by estimating an appropriate transition probability matrix for $\{x_t^N\}$. Suppose that we pretend, temporarily, that $\{y_t\}$ actually is hidden Markov of this form, and proceed to estimate the corresponding $m \in M_N$ transition matrix for $\{x^N\}$. Let $\hat{m}_{N,n}$ be the maximum-likelihood estimator, given $y_1, y_2, ...y_n$, and let $\hat{y}_t^{N,n}$ be the corresponding hidden Markov model. We have shown that for essentially arbitrary ergodic processes $\{y_t\}$, $\{\hat{y}_t^{N,n}\} \to \{y_t\}$, provided that $N = N_n \uparrow \infty$ sufficiently slowly.

The conclusion is that we can, at least in principle, fit essentially arbitrary one-dimensional stationary processes with hidden Markov models. The corresponding result in higher dimensions appears to be substantially more difficult, although the above-mentioned result about approximation with Markov random fields strongly suggests that an estimation result is also available.

We have used the one-dimensional result to fit some speech waveforms with excellent results. We have also done some preliminary experiments with some simple textures, under the presumption that the estimation result in higher dimensions is true as well, and have obtained good renderings of these simple patterns.

4. TWO AND THREE DIMENSIONAL GLOBAL SHAPE MODELS.

A distinctive feature of the nonrigid object recognition problems and models in §3 is their one-dimensional structure; the graphs describing connectivity of the objects are *linear* or branching linear graphs. The one-dimensional structure is used to full advantage when we can apply the principle of optimality and dynamic programming to compute solutions of global optimization problems.

The basic theme in §3 of using deformable templates, modeled probabilistically, has been explored in a variety of settings, including higher dimensional ones for describing surfaces and volumes. It is useful to consider the deformable template paradigm in some generality, to identify common features of the mathematical models and problems, and to develop approaches and algorithms which may transcend a specific motivating application.

Many image ensembles are characterized both by exhibiting characteristic shape and by a high degree of variability. For example, medical imaging usually leads to very structured pictures whose appearance expresses anatomical, histological, or cytological properties. On the other hand they vary a lot, not only between individuals, but also for one individual, from one recording to another. As a matter of fact the variability is crucial since only understanding of what normal variation means makes it possible to define precisely and to

detect pathologies.

For man-made objects variability will also be important if the objects have many degrees of freedom and the corresponding parameters are not known in advance. This can be the case even for rigid objects: for instance, in infrared imagery, thermal signatures may be unknown a priori and have many degrees of freedom.

Purely local methods, such as noise suppression and edge detection, have shown their usefulness, especially for texture patterns, but are not sufficiently powerful when the characteristic shapes are of global nature. For this reason we have introduced and applied global shape models based on deformable templates to a number of image ensembles. We refer to a study of leaf shape, Knoerr [37], human hands pictures acquired by visible light cameras, Grenander-Chow-Keenan [30] and by X-ray cameras, Amit-Grenander-Piccioni [3], and range data images, Grenander-Keenan [31]. Others are under way, including the work on arteriograms described above and ongoing research on recognition of shapes of mitochondria, in collaboration with Michael Miller at Washington University. See Table 4.1.

Collaborators	*Application*	*Space Dimension*	*Object Dimension*	*Imaging Modality*
Chow–Grenander–Keenan	HANDS	2D	1D Template	Visible Light
Knoerr	LEAVES	2D	1D Template	Visible Light
Amit–Grenander–Piccioni	HAND XRAYS	2D	2D Template	X-rays
Grenander–Keenan	RANGE DATA	3D	2D Template	Laser Radar
Grenander–Miller	MITOCHONDRIA	2D	1D Template Multiple Objects	Electron Microscopy

Table 4.1. Studies developing global shape models

4.1 The basic idea is to incorporate '*typical shape*' into one or several templates, while

variability is expressed by a (prior) probability distribution. The template is deformed by applying a group of transformations and the resulting image is then observed via some acquisition technology, usually with observational noise.

We then make inferences about the true image (not corrupted by noise) using the knowledge contained in its (posterior) probability distribution conditioned by having viewed the observed one.

The inference algorithms obtained are, if the global shape model is correct, the best possible ones. They do not aim just for optimal restoration of the image but for a *structured analysis*, image understanding, in the following sense.

Intrinsic understanding: Assuming the global shape model used to be true the algorithm should give objective information in quantitative form about image features such as lengths, areas of parts of the image, outline of these parts, description of the topology linking these parts.

Extrinsic understanding: The algorithm should signal abnormalities not consistent with the (normal) variation expressed by the global shape model, and flag the suspicious locations in the picture.

4.2 Mathematically this research strategy is formalized in terms of a generator space G of primitives, for example line segments, conic arcs, surface stars, etc. (choices of $G, S, \cdots$ and so on in our earlier studies are shown in Table 4.2). A generator $g \in G$ has a number $\omega(g)$ of bonds attached to it, $\beta_1, \beta_2, \cdots \beta_{\omega(g)}$ which are used to define locally regular configurations

$$c = \sigma(g_1, g_2, \cdots g_n) \tag{4.1}$$

where σ is a graph from a graph family Σ, the connection type that expresses global regularity, and the g_i in (4.1) are situated at the sites of the graph σ. Two sites i_1 and i_2 that are connected by σ should satisfy a local regularity condition

$$\beta_{j_1}(g_{i_1}) \rho \beta_{j_2}(g_{i_2}) = \mathbf{TRUE},$$

where ρ is a binary truth-valued function. For instance, in the spline representation of a handwritten digit "5", generators g are oriented line segments (vectors), the graph structure σ is linear, and the local regularity condition ρ enforces connectivity/continuity of successive segments.

The configurations thus obtained form the configuration space $\mathcal{C}(\mathcal{R})$, where the regularity $\mathcal{R}$ is

$$\mathcal{R} = < \rho, \Sigma > .$$

A template configuration $c_{temp} = \sigma(g_1^{(0)}, g_2^{(0)}, \cdots g_n^{(0)}) \in \mathcal{C}(\mathcal{R})$ is subjected to deformations

$$c_{temp} \to c = \sigma(s_1 g_1^{(0)}, s_2 g_2^{(0)}, \cdots s_n g_n^{(0)})$$

where the s_i are elements of a group S of transformations (similarities) $S : G \to G$.

To account for variability around the template(s) we introduce a probability distribution on S^n by a density

$$\frac{1}{Z} \prod_{\sigma} A[\beta_{j_1}(g_{i_1}), \beta_{j_2}(g_{i_2})]$$

with some acceptor function $A(\cdot, \cdot)$, similarly to the definitions of Markov random fields in statistical physics.

This induces a probability density π, the prior, on the configuration space $\mathcal{C}(\mathcal{R})$. For a given (noisy) image acquisition technology we shall let $L(\mathcal{I}^{\mathcal{D}}|c)$, the likelihood function, express the conditional probability density of observing the deformed image $\mathcal{I}^{\mathcal{D}}$ if c is the true (and unknown) configuration.

Bayes theorem then gives us the posterior density

$$p(c|\mathcal{I}^{\mathcal{D}}) \propto \pi(c) L(\mathcal{I}^{\mathcal{D}}|c) \tag{4.2}$$

and it remains to simulate (4.2); if we can do this by a computationally feasible algorithm the optimal image inference falls out almost automatically.

To exemplify the above, somewhat terse summary, the presentation in Table 4.2 shows how $G, S, \cdots$ have been chosen in earlier studies. Complete details can be found in the references.

4.3 The simulation of (4.2) is not straightforward since the domain of $p(\cdot|\mathcal{I}^{\mathcal{D}})$ is a complicated space.

In the first attempts, Grenander-Chow-Keenan [30] in 2D, Grenander-Keenan [31] in 3D, simulation of (4.2) was achieved by stochastic relaxation described in Grenander [29], Geman-Geman [18]. While this was carried out with success it also became clear that as we go ahead to more detailed shape models it is difficult to develop well structured code for stochastic relaxation. The main reason is that when we use multi-stage algorithms with stacked similarity groups (see below) the data structures tend to be messy.

To avoid this our more recent computer implementation have used diffusion methods, also described in Geman-Geman [18] and Grenander [29]. This is organized as follows.

Application	*Generators G*	*Transformations S*	*Graph Structures* Σ
HANDS	Vectors	$US(2) \times O(2)$	CYCLIC
LEAVES	Vectors	$US(2) \times O(2)$	CYCLIC
XRAYS	Points (with old values)	Translation Group in $\mathbb{R}^2$	LATTICE
RANGE DATA	Facets of Polyhedra	$GL(3)$	TESSELATED ICOSAHEDRON
MITOCHONDRIA	Vectors	$US(2) \times O(2)$ and $O(2)$	MULTIPLE CYCLES

Table 4.2. Examples of Generators, Groups and Graphs

In all cases of interest the group S has been a low dimensional Lie group. Introduce a stochastic differential equation (S.D.E.) for the group element $z = (s_1, s_2, \cdots s_n) \in S^n$

$$dz(t) = f[z(t)]dt + dW(t) \tag{4.3}$$

In (4.3) W is a Wiener process taking values in S^n, which is locally Euclidean so that (4.3) can be made meaningful. The function $f(\cdot)$ is the gradient of the energy function $H(\cdot)$ with

$$p(c|\mathcal{I}^{\mathcal{D}}) = \exp\ H(c) \tag{8}$$

Initializing $z(0)$ and solving (4.3) iteratively we go on until t is big enough to make the distribution of $z(t)$ sufficiently close to the equilibrium distribution, where this has to be made rigorous by arguments from the t[illegible] of S.D.E.'s. We actually do this in stages, starting in low dimensional subgroups of S^n, stacking the subgroups, and then slowly let the subgroup increase; a detailed analysis of how this should be done can be found in Amit-Piccioni [4]. The subgroups are obtained from the subspaces in the Lie algebra associated with S^n.

While our computer experiments employing stochastic relaxation required considerable computing power and were executed on an IBM 3090 and a CRAY XMP, our more

recent experiments using diffusion simulation could be done on a SUN workstation and even on a 386 PC.

4.4 If the number of objects is unknown to begin with the model must be extended to prior densities π defined on a configuration space $\mathcal{C}(\mathcal{R}_{mult})$ where the regularity $\mathcal{R}_{mult} = < \rho, \Sigma_{mult} >$ is over a larger connection type

$$\Sigma_{mult} = \bigcup_{k=0}^{\infty} \Sigma^k; \; \Sigma^k = \underbrace{\Sigma \times \Sigma \times \cdots \times \Sigma}_{k \text{ times}}.$$

In order that the algorithm be able to understand such images it must be able to create (and annihilate) hypotheses about objects. But

$$\mathcal{C}(\mathcal{R}_{mult}) = \bigcup_{k=0}^{\infty} \mathcal{C}(< \rho, \Sigma^k >) \tag{10}$$

is made up of a denumerable sequence of continua so that instead of the S.D.E. in (4.3) the dynamics must be able to jump from one continuum to another in addition to more continuously (changing shape) within one of them.

Therefore (4.3) is replaced by a jump-diffusion process. This concept is old, going back to the 1930s but has not been applied to hypothesis generation as above and we plan to explore this possibility in depth and apply it to the acquisition technologies mentioned.

Two illustrative applications now under study are: finding mitochondria as well as membranes in electron microscopy with high magnification; and structured restoration in 3D, in particular images obtained by optical sectioning and by range finding laser radar.

5. THREE DIMENSIONAL SHAPE RECONSTRUCTION

The problem of estimating or reconstructing geometric properties of 3D surfaces, such as orientation, height (depth), curvature, or determining topographic maps, from digital image intensities, is known as "shape from shading" in computer vision, and "radarclinometry" in airborne or spaceborne imaging (synthetic aperture) radar systems. In most applications, the estimation of 3D geometric properties is coupled with the determination of surface's composition, e.g. albedo or dielectric properties. A typical method for estimating 3D shapes or topographic maps, is "stereo" based on a stereo pair of optical photographs [32,41] or radar images [14]. However, stereo vision or "stereopsis" (which is the single most important process by which human beings obtain depth information) has played a limited role in computer vision due to its computational demands and lack of accuracy. In the case of spaceborne radars (such as the Magellan Synthetic Aperture Radar which will map the surface of Venus) a pair of stereo images is typically unavailable.

The 3D shape reconstruction problem from a single image is underdetermined ("ill-posed") due to the loss of information in passing from a 3D continuous physical scene to sampled, quantized, and typically degraded by blur noise and radiometric distortions, 2D arrays. Most approaches [33,34,44] to the shape-from-shading problem, involve preprocessing steps which encompass removal of noise and other aspects of degredation, and segmentation and boundary detection (e.g. occluding boundaries). In [26,48] a coherent Bayesian-Markov Random Fields procedure for estimating 3D geometric properties and surface composition (albedo) from a single image with or without degradation, has been developed. The procedure has been tested successfully on video images, and we are currently experimenting with simulated synthetic aperture radar (SAR) data. The methodology accommodates the use of multiple data, e.g. multiple wavelength images, or images from different sensors.

The basic strategy is the same for video and SAR data, but the two cases are technically different primarily because of differences in the data acquisition processes and data interpretation. In the case of optical images, the procedure is formulated as follows: Let $S^P = \{i = (i_1, i_2) : 1 \leq i_1, i_2 \leq N\}$ be the pixel lattice (image grid). The "ideal" (undegraded) image intensities $X^P = \{X_i : i \in S^P\}$ are related to the geometry of a surface via the "irradiance equation" [33]

$$X_i^P = R(\vec{N}_i, \vec{S}, \vec{V}, \rho_i), i \in S^P \tag{5.1}$$

where $\vec{N}_i$ is the surface unit normal at a physical point corresponding to pixel $i \in S^P$, $\vec{S}$ is the direction of the illumination source (for an extended source, $\vec{S}$ is an "effective"

direction), $\vec{V}$ is the direction of the camera (which, for simplicity, is assumed to be constant throughout the image, i.e. we assume orthographic projections), $\rho = \{\rho_i : i \in S^P\}$ is the albedo function, and R is the *reflectance map* which has been studied extensively for various illumination sources and material. The actual observed (recorded) data $Y = \{Y_i : i \in S^P\}$ are a degraded version of X^P given by a transformation typically of the form

$$Y_i = \psi\{\phi[(KX^P)_i], \eta_i\} \tag{5.2}$$

where K accounts for blur ("point spread function"), ϕ accounts for radiometric distortions, η_i is a collection of noise processes, and ψ defines the noise mechanism(s) (e.g. additive, multiplicative, etc).

The target of estimation is the surface elevation ($\equiv$ depth $\equiv$ height) $z = \{z_i : i \in S^P\}$ which is a discrete version of the surface function $z = z(u), u = (u_1, u_2) \in I\!R^2$ above the image plane. Often, the albedo $\rho = \{\rho_i\}$ and the light source direction $\vec{S}$ are unknown and need to be estimated simultaneously with z. The depth $z = \{z_i\}$ can be recovered from an estimation of the unit normals $N = \{\vec{N}_i : i \in S\}$ provided that the estimated normals satisfy the *integrability condition* which corresponds to a discrete version of $z_{u_1,u_2} = z_{u_2,u_1}$.

The procedure for estimating $N = \{\vec{N}_i\}$ (and hence depth) is based on stochastic regularization [17,23,29] via Gibbs distributions designed to capture our a priori expectations about surfaces: surfaces are locally smooth (and hence, $N = \{\vec{N}_i\}$ and curvature change locally smoothly), while orientation jumps along surface discontinuities or occluding boundaries. In addition to the process $N = \{\vec{N}_i : i \in S^P\}$, we introduce an auxiliary "edge" process X^E as in previous work [17,23] on restoration. The process X^E is indexed by the dual lattice S^E of S^P, i.e. $X^E = \{X^E_t : t \in S^E\} = \{X^E_{<ij>} : i,j \in S^P, |i-j| = 1\}$, and a "site" $t =< ij > \in S^E$ corresponds to a putative edge between pixels i and j. It is a binary process with $X^E_t = 1$ (resp. 0) indicating presence (resp. absence) of an edge at $t \in S^P$.

The joint process (N, x^E) is chosen to be a Markov Random Field (MRF) with distribution (the *prior*)

$$\pi(N, x^E) = \frac{1}{Z} e^{-U(N,x^E)}$$

where the energy function $U(N, x^E)$ consists of two terms

$$U(N, x^E) = U_1(N, x^E) + U_2(x^E) \tag{5.3}$$

U_1 reflects our expectations about interactions between normals and edges, while U_2 reflects boundary "organization". Both terms are constructed in terms of "local interactions"

corresponding to some neighborhood system. More specifically, in our experiments U_1 was chosen to be

$$U_1(N, x^E) = \theta_1 \sum_{<ij>} (1 - x^E_{<ij>}) \phi(|\vec{N}_i - \vec{N}_j|)$$

$$+\theta_2 \sum_{[i,j]} (1 - \sigma^E_{<ij>}) \phi(|\vec{N}_i - \vec{N}_j|) \tag{5.4}$$

where $\theta_1, \theta_2 > 0, [i,j]$ denotes nearest-neighbor diagonals (i.e. $|i-j| = \sqrt{2}$), and

$$\sigma^E_{[i,j]} = \begin{cases} 1 \ , & \text{if} \quad x^E_{t_1} x^E_{t_2} = 1 \quad \text{or} \quad x^E_{t_1} x^E_{t_3} = 1 \\ & \text{or} \quad x^E_{t_2} x^E_{t_4} = 1 \quad \text{or} \quad x^E_{t_3} x^E_{t_4} = 1 \\ 0 \ , & \text{otherwise} \end{cases}$$

where i, j, t_1, t_2, t_3, t_4 are as in Figure 5.1 or in its rotation by $\pi/2$.

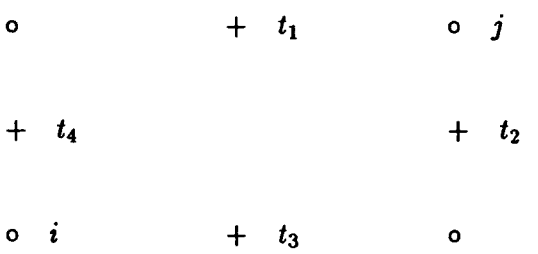


Figure 5.1.

The function $\phi(\cdot)$ was chosen to be $\phi(|\vec{N}_i - \vec{N}_j|) = -1 + \frac{1}{2}|\vec{N}_i - \vec{N}_j|^2 = -\vec{N}_i \cdot \vec{N}_j$. Because of the constraint $|\vec{N}_i| = 1$, the prior resulting from this choice of ϕ is non-Gaussian even if $\theta_2 = 0$ and $x^E \equiv 0$. In fact, model (5.3) has worked well [48] in some cases even without the edge process x^E.

The term $U_2(x^E)$ is designed to capture our generic expectations about discontinuities (and occluding boundaries): most physical points are away from discontinuities; discontinuities are usually persistent (no isolated or abandoned segments); discontinuity intersections, sharp turns, etc, are relatively unlikely. For specific choices of U_2 we refer to [17, §4.3]. Some of the above regularity properties may also be captured by "penalty functions" [20], a procedure which may be useful for textured surfaces.

In addition to the data (5.1) or (5.2), and the prior induced by (5.3), we have a deterministic constraint $V_1(N) = 0$ (see [26]) corresponding to the integrability condition. Assuming that $\vec{S}$ and ρ are known, the estimation of $N = \{N_i\}$ is reduced to the following constrained optimization problems: In the case of observable X^P (i.e. no degradation), we minimize (5.3) subject to the constraints $V_1(N) + V_2(N) = 0$ where $V_2(N) = \sum_{i \in S^P} |x^P_i -$

$R(\vec{N}_i)|^2$. In the presence of degradation, (5.2) induces a conditional probability $P(Y|N) = P(Y|X^P)$. This together with the prior yield the posterior distribution $P(N|Y)$. Then N is estimated by maximizing $P(N|Y)$ subject to the constraint $V_1(N) = 0$. This procedure can be extended [48] to estimate also $\vec{S}$ and ρ when ρ is constant through the image.

Figure 5.2 shows an experiment with an egg imaged under uncontrolled illumination (using a desk lamp). The surface of the egg was assumed matte. No degredation was considered. The algorithm—a combination of constrained annealing [18] and Iterated Conditional Mode (ICM) [7]—estimated in addition to N, the albedo ρ and an effective light source direction $\vec{S}$: (a) original image 64×64, (b) image reconstructed from the estimated $N, \vec{S}$, and ρ, (c) reconstructed egg illuminated from the x-direction, (d) reconstructed egg illuminated from the y-direction.

6. TEXTURE ANALYSIS

Texture is a dominant feature in remote sensing and other data, and texture discrimination is important in many applications including: determining the spatial distribution, size and shapes of ice floes in the ocean; monitoring polar ice cover; analyzing satellite data for resource classification, crop assessment, weather prediction, and geologic mapping; industrial quality control as, for example, in the inspection of silicon wafers where low magnification views of memory arrays appear as highly structured textures; inferring 3D geometric properties of surfaces ("shape-from-texture").

Statistical methods for texture discrimination in remote sensing are prevalent [17,46]. However, until recently most techniques employed conventional methods such as principal component analysis, in which pixels are classified individually and independently. More recently [11,12,13,16] there has been an increasing emphasis on spatial methods (based on Markov and other random fields) which reflect the presence of spatial coherence: e.g. crops or vegetation are expected to grow in large locally random but "globally" homogeneous regions.

In [19,20,22], we have developed three complimentary procedures for texture discrimination based on "label models" which involve two coupled processes: the grey-level intensity process and the label process. In the first procedure [19,22] the coupled process is a two-tiered MRF : one level for the label (or region) process, a simple Ising type process, and the other level for the intensity process with a specified distribution conditional on the region labels. These conditional distributions (also MRF) are the "models" of specific textures. These are properly designed parametric models whose parameters are estimated from one

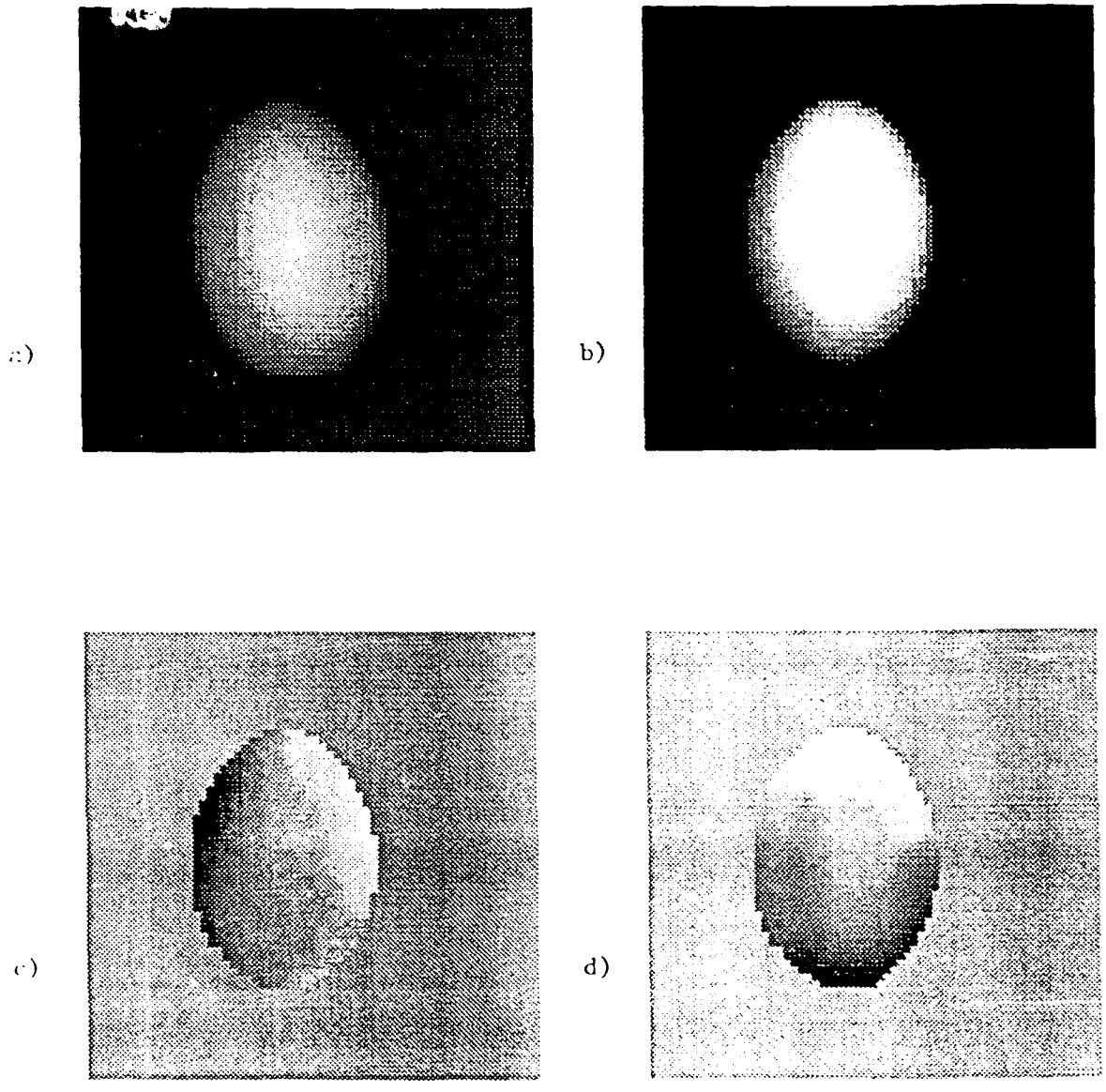


Figure 5.2. Egg images. (a) Original; (b) Reconstructed from estimated shape information; (c) Reconstructed, illuminated from the right; (d) Reconstructed, illuminated from above.

or more actual texture samples. This discrimination procedure is dedicated to a fixed repertoire of textures. The second and third procedures [20], regard texture discrimination as partitioning and boundary detection, respectively. The partition labels do not *classify* (as in the first procedure), but they are *generic* and are assigned to pixels or blocks of pixels with block size ("label resolution") depending on the resolution of the data and intended interpretation; the boundary labels are just "on" and "off" and are associated with an interpixel sublattice. In contrast to the first procedure, the latter two involve no "modeling" of

textures. Partitioning and boundary placements are driven by spatial statistics features selected in terms of the Kolmogorov-Smirnov distance applied either to raw data ("first-order statistics") or to transformed data corresponding to higher order statistics (e.g. window means, range, variance, "directional residuals", etc); they also involve "hard constraints" which penalize unwanted or "forbidden" label configurations.

Figures 6.1–6.3 show texture discrimination experiments with the first, second, and third procedure, respectively. Figure 6.1 involves four textures: wood, carpet, cloth, and plastic; the left panel is the textured scene, and the right panel shows the segmentation with texture labels coded by grey bevel. Figure 6.2 is SAR image of ice floes in the ocean; three partition labels were used, one for water and two (dark and light) for ice: (a) original image 512 × 512, (b) shows the evolution of stochastic relaxation; the upper left panel is the random starting configuration and the bottom right is the final configuration. The original for Figure 6.3 is the same as in Figure 6.2; the Figure shows sixteen "snapshots" of stochastic relaxation for the boundary model—every third sweep from a sequence of sweeps.

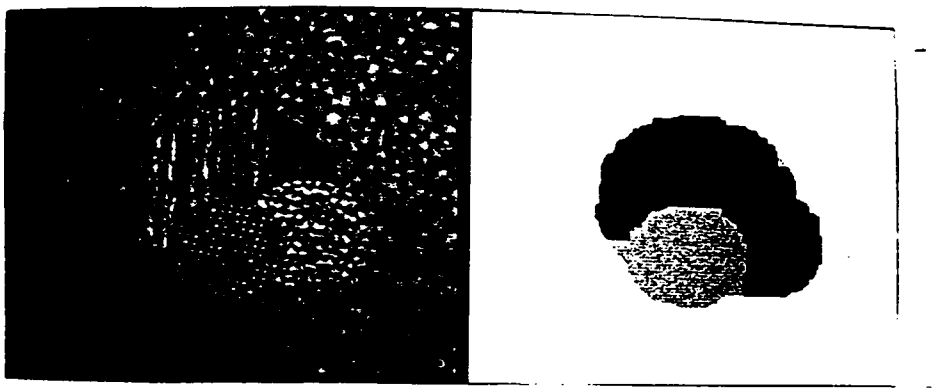


Figure 6.1. Segmentation of four textures; wood, carpet, and cloth on a plastic background

These models are adequate for texture discrimination, but not for texture synthesis. For the purpose of texture synthesis, we have explored (see [1], [2]) the following class of MRF models:

$$U(x) = \frac{1}{2}\sum_{i,j\in S} A(i-j)x_i x_j + \sum_{i\in S} p(x_i;\lambda), x_i \in I\!R \tag{6.1}$$

where A is a positive definite matrix with $A(i) = A(-i)$, and $p(\xi,\lambda), \xi \in I\!R$ is a polynomial, $p(\xi,\lambda) = \lambda\xi + \lambda_2\xi^2 + \cdots + \lambda_{2M}\xi^{2M}$, of even degree with $\lambda_{2M} > 0$. Figure 6.4 shows a wood like texture generated from a MRF corresponding to (6.1) with a quartic polynomial

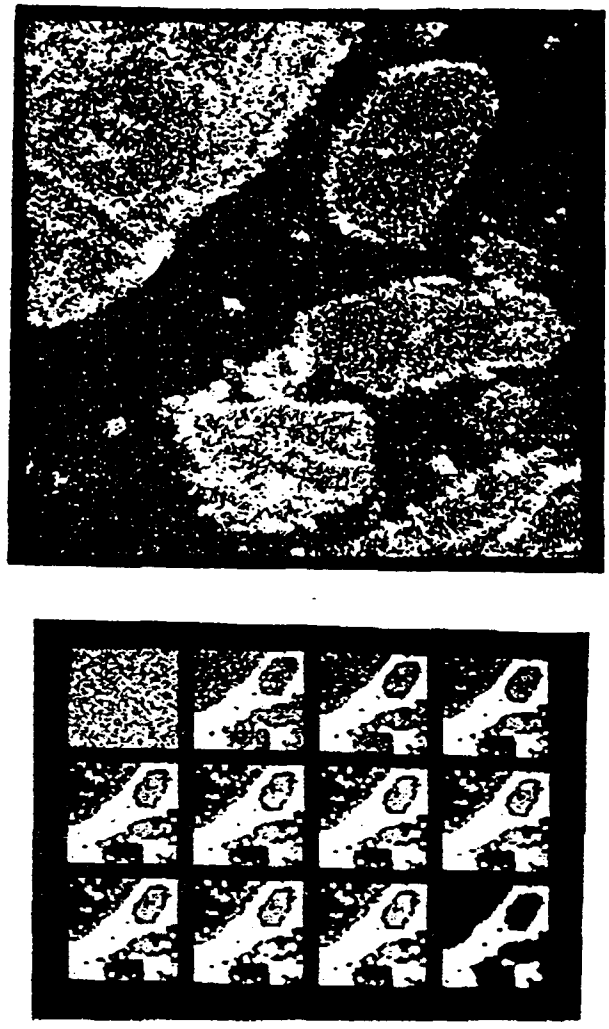


Figure 6.2. Segmentation of SAR image of ice floes, partition model

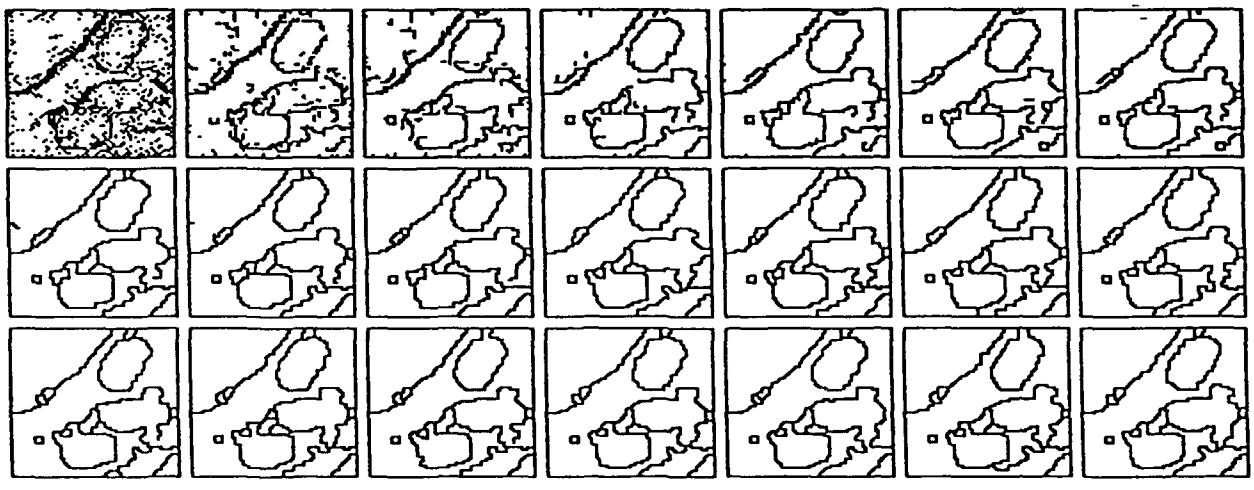


Figure 6.3. Segmentation of SAR image of ice floes, boundary model

$p(\xi;\lambda)$, and $A(i-j) = 0$ for $|i-j| > 2$ (the model had exactly nine parameters). The first term in (6.1) corresponds to a Gaussian distribution. Gaussian distributions have been used successfully for generating textures in [9]. A large class of models of the form (6.1) can be defined [1] at all levels of resolution including the continuum. This may be useful for scale and rotation invariant segmentation of textures.

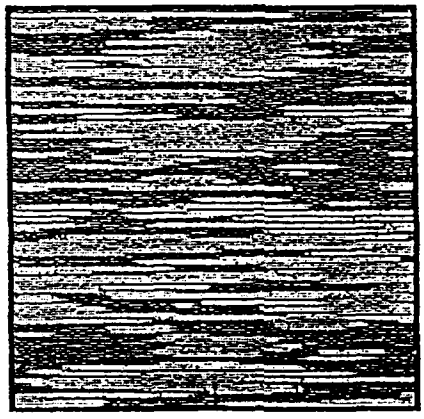


Figure 6.4. Wood-like texture generated by MRF

BIBLIOGRAPHY

1. M. Almeida, "Statistical inference for MRF with unbounded continuous spins and applications to texture representation," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1989.

2. M. Almeida and B. Gidas, "A variational method for estimating the parameters of MRF from complete or incomplete data," preprint 1989, Brown University.

3. Y. Amit, U. Grenander, and M. Piccioni, "X-Rays," Submitted for publication, 1989.

4. Y. Amit and M. Piccioni, "A multilevel Markov process for the estimation of Gaussian random fields with nonlinear observations," Technical Report, Division of Applied Mathematics, Brown University, 1989.

5. L.R. Bahl, F. Jelinek and R.L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE PAMI*, 5, 179-190, 1983.

6. R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing* 46, 1-21, 1989.

7. J. Besag, "On the statistical analysis of dirty pictures" (with discussion), *J. Royal Stat. Soc.* Ser. B 48, 259-302, 1986.

8. D.J. Burr, "A dynamic model for image registration," *Computer Graphics and Image Processing*, 15, 102-112, 1981.

9. F. Cohen, "Markov random fields for image modelling and analysis", preprint 1988.

10. D.B. Cooper, H. Elliott, F. Cohen, L. Reiss and P. Symosek, "Stochastic boundary estimation and object recognition," *Computer Graphics and Image Processing*, 12, 326-356, 1980.

11. G.R. Cross and A.K. Jain, "Markov random field texture models," *IEEE PAMI* 5, 25-39, 1983.

12. H. Derin and W.S. Cole, "Segmentation of textured images using Gibbs random fields," *Comput. Vis. Graph. and Image Process.*, 35 72-98, 1986.

13. H. Derin and H. Elliott, "Modelling and segmentation of noisy and textured images using Gibbs random fields," *IEEE PAMI*, 9 39-55, 1987.

14. C. Elachi, *Spaceborne Radar Remote Sensing-Applications and Techniques*, IEEE Press, 1988.

15. M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, 22, 67-92, 1973.

16. A. Gagalowitz and S. D. Ma, "Sequential synthesis of natural textures," *Computer Vision, Graphics and Image Processing*, 30, 289-315, 1985.

17. D. Geman, "Random fields and inverse problems in imaging," *Lecture Notes in Mathematics*, Springer–Verlag, to appear 1990.

18. D. Geman and S. Geman, "Relaxation and annealing with constraints", Complex Systems Tech. Report No. 35, Division of Applied Mathematics, Brown University, 1987.

19. D. Geman, S. Geman and C. Graffigne, "Locating texture and object boundaries," in *Pattern Recognition Theory and Applications*, eds. P.A. Devijer and J. Kittler, Springer–Verlag, 1987.

20. D. Geman, S. Geman, C. Graffigne and P. Dong, "Boundary detection by constrained optimization," *IEEE-PAMI*, 12, 609-628, 1990.

21. D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," preprint 1990, Department of Math. and Stat., University of Massachusetts.

22. S. Geman and C. Graffigne, "Markov random field image models and their applications to computer vision," *Proceedings of the International Congress of Mathematicians, 1986.*, ed. A.M. Gleason, American Mathematical Society, Providence, 1987.

23. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE-PAMI* 6, 721-741, 1984.

24. S. Geman and D.E. McClure, "Statistical methods for tomographic image reconstruction," *Proceedings of the 46th Session of the International Statistical Institute,* Bulletin of the ISI, Vol. 52, 1987.

25. B. Gidas, "A renormalization group approach to image processing problems," *IEEE-PAMI* 11, 164–180, 1989.

26. B. Gidas and J. Torreão, "A Bayesian/geometric framework for reconstructing 3-D shapes in robot vision," SPIE, *High Speed Computing II*, Vol. 1058, 86-93, 1989.

27. U. Grenander, "A unified approach to pattern analysis," *Advances in Computers*, Vol. 10, Academic Press, New York, 1970.

28. U. Grenander, *Abstract Inference*, John Wiley & Sons, Inc., New York, 1981.

29. U. Grenander, "Tutorial in Pattern Theory," Division of Applied Mathematics, Brown University, Technical Report, 1983.

30. U. Grenander, Y.S. Chow and D. Keenan, *HANDS, A Pattern Theoretic Study of Biological Shapes*, Springer-Verlag, New York, 1988 (in press).

31. U. Grenander and D. Keenan, "A computer experiment in pattern theory," *Commun. Statist. Stochastic Models*, Vol. 5(4), 532–553, 1989.

32. W.E.L. Grimson, "A computational theory of visual surface interpolation," *Phil. Trans. R. Soc. London*, B298, 399-427, 1982.

33. B.K.P. Horn, *Robot Vision*, M.I.T. Press, 1986.

34. B.K.P. Horn, "Height and gradient from shading," M.I.T., A.I. Memo No 1105, 1989.

35. D.A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, 40, 1098-1101, 1952.

36. D. Kandel, E. Domany, D. Ron, A. Brandt, and E. Loh, Jr., "Simulation without critical slowing down," *Phys. Rev. Letters* 60 1591-1594, 1988.

37. A. Knoerr, "Global models of natural boundaries: theory and applications," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1988.

38. E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

39. K.F. Lee, "Large vocabulary speaker independent continuous speech recognition: The SPHINX system," Ph.D. dissertation, Computer Science, Carnegie Mellon University, 1988.

40. K.M. Manbeck, "Bayesian statistical methods applied to emission tomography with physical phantom and patient data," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1990.

41. D. Marr, *Vision*, W. H. Freeman 1982.

42. D.E. McClure and S. Shwartz, "A method of image representation based on bivariate splines," Center for Intelligent Control Systems Report CICS–P–113, submitted to *IEEE PAMI*, 1989.

43. J.A. Mertus, "Self calibrating methods for image reconstruction in emission computed tomography," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1988.

44. T. Poggio, V. Torro, and C. Koch, "Computational vision and regularization theory", *Nature* 317, 314-319, 1985.

45. L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, 77, 257-286, 1989.

46. B.D. Ripley, "Statistics, images, and pattern recognition," *Canadian J. Statistics*, 14, 83-111, 1986.

47. A.H. Swendsen and J.S. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Phys. Rev. Letters*, 58, 86-88, 1987.

48. J. Toreão, "A Bayesian approach to 3-D shape estimation for robot vision," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1989.

49. B. Widrow, "The rubber mask technique, Part I," *Pattern Recognition*, 5, 175-211, 1973.